

LAPORAN MILESTONE 2 TUGAS BESAR IF 1210



Disusun Oleh : Kelompok L - K04

Francis Galton	18224072
Rafael Sean Hadipranoto	18224100
Rashid May	18222014
Derren Christopher Boenardi	18224040
Sri Laksmi Purwoningtyas	18221009
Geodipa Afatha Ryu M.F.Z.	18224124

MATA KULIAH ALGORITMA PEMROGRAMAN 1
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA - KOMPUTASI
INSTITUT TEKNOLOGI BANDUNG
2025

HALAMAN PERNYATAAN

Kami,

Francis Galton	18224072
Rafael Sean Hadipranoto	18224100
Rashid May	18222014
Derren Christopher Boenardi	18224040
Sri Laksmi Purwoningtyas	18221009
Geodipa Afatha Ryu M.F.Z.	18224124

"Kami menyatakan bahwa kami mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui kami mengerjakan tugas besar ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Algoritma dan Pemrograman 1 Semester 2 2024/2025."

:

DAFTAR ISI

HALAMAN PERNYATAAN.....	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	6
DAFTAR GAMBAR.....	7
DESKRIPSI PERSOALAN.....	9
RENCANA IMPLEMENTASI.....	10
PEMBAGIAN KERJA.....	20
HASIL Pengerjaan.....	24
DESIGN COMMAND.....	26
1. F01 - Login.....	26
2. F02 - Register.....	26
3. F03 - Logout.....	26
4. F04 - Lupa Password.....	26
5. F05 - Menu & Help.....	26
6. F06 - Denah Rumah Sakit.....	26
7. F07 - Lihat User.....	27
8. F08 - Cari User.....	27
9. F09 - Lihat Antrian.....	27
10. F10 - Tambah Dokter.....	27
11. F11 - Diagnosis.....	27
12. F12 - Ngobatin.....	27
13. F13 - Aku boleh pulang ga, dok?.....	28
14. F14 - Daftar Check-Up.....	28
15. F15 - Antrian Saya!.....	28
16. F16 - Minum Obat.....	28
17. F17 - Minum Penawar.....	28
18. F18 - Exit.....	28
DESIGN KAMUS DATA.....	30
1. F01 - Login.....	30
2. F02 - Register.....	30
3. F03 - Logout.....	30
4. F04 - Lupa Password.....	30
5. F05 - Menu & Help.....	30
6. F06 - Denah Rumah Sakit.....	30
7. F07 - Lihat User.....	31

8. F08 - Cari User.....	31
9. F09 - Lihat Antrian.....	31
10. F10 - Tambah Dokter.....	31
11. F11 - Diagnosis.....	32
12. F12 - Ngobatin.....	32
13. F13 - Aku boleh pulang ga, dok?.....	32
14. F14 - Daftar Check-Up.....	33
15. F15 - Antrian Saya!.....	33
16. F16 - Minum Obat.....	33
17. F17 - Minum Penawar.....	34
18. F18 - Exit.....	34
DESIGN DEKOMPOSISI.....	35
1. F01 - Login.....	35
2. F02 - Register.....	35
3. F03 - Logout.....	36
4. F04 - Lupa Password.....	36
5. F05 - Menu & Help.....	37
6. F06 - Denah Rumah Sakit.....	37
7. F07 - Lihat User.....	38
8. F08 - Cari User.....	39
9. F09 - Lihat Antrian.....	40
10. F10 - Tambah Dokter.....	41
11. F11 - Diagnosis.....	41
12. F12 - Ngobatin.....	42
13. F13 - Aku boleh pulang ga, dok?.....	42
14. F14 - Daftar Check-Up.....	43
15. F15 - Antrian Saya!.....	44
16. F16 - Minum Obat.....	45
17. F17 - Minum Penawar.....	46
18. F18 - Exit.....	46
SPESIFIKASI MODUL.....	47
1. Main Program.....	47
2. F01 - Login.....	49
3. F02 - Register.....	50
4. F03 - Logout.....	51
5. F04 - Lupa Password.....	51
6. F05 - Menu & Help.....	53
7. F06 - Denah Rumah Sakit.....	54

8. F07 - Lihat User.....	55
9. F08 - Cari User.....	56
10. F09 - Lihat Antrian.....	58
11. F10 - Tambah Dokter.....	60
12. F11 - Diagnosis.....	63
13. F12 - Ngobatin.....	66
14. F13 - Aku boleh pulang ga, dok?.....	69
15. F14 - Daftar Check-Up.....	70
16. F15 - Antrian Saya!.....	73
17. F16 - Minum Obat.....	74
18. F17 - Minum Penawar.....	75
19. F18 - Exit.....	76
HASIL PENGUJIAN PROGRAM.....	77
1. Inisialisasi Main Program.....	77
2. F01 - Login.....	77
3. F02 - Register.....	79
4. F03 - Logout.....	80
5. F04 - Lupa Password.....	80
6. F05 - Menu Help.....	81
7. F06 - Lihat Denah.....	83
8. F07 - Lihat User.....	84
9. F08 - Cari User.....	85
10. F09 - Lihat Antrian.....	86
11. F10 - Tambah Dokter.....	86
12. F11 - Diagnosis.....	88
13. F12 - Ngobatin.....	90
14. F13 - Aku boleh pulang ga, dok?.....	90
15. F14 - Daftar Check-Up.....	91
16. F15 - Antrian Saya!.....	92
17. F16 - Minum Obat.....	93
18. F17 - Minum Penawar.....	94
19. F18 - Exit.....	95
20. B03 - Aura.....	95
LAMPIRAN.....	97

DAFTAR TABEL

Tabel 1.1 Rencana Implementasi.....	10
Tabel 2.1 Pembagian Kerja.....	20
Tabel 3.1 Hasil Pengerjaan.....	24

DAFTAR GAMBAR

Gambar 6.1 Flowchart Login.....	35
Gambar 6.2 Flowchart Register.....	35
Gambar 6.3 Flowchart Logout.....	36
Gambar 6.4 Flowchart Lupa Password.....	36
Gambar 6.5 Flowchart Menu & Help.....	37
Gambar 6.6 Flowchart Lihat Denah.....	37
Gambar 6.7 Flowchart Lihat Ruangan.....	38
Gambar 6.8 Flowchart Lihat User.....	38
Gambar 6.9 Flowchart Cari User.....	39
Gambar 6.10 Flowchart Lihat Antrian.....	40
Gambar 6.11 Flowchart Tambah Dokter.....	41
Gambar 6.12 Flowchart Diagnosis.....	41
Gambar 6.13 Flowchart Ngobatin.....	42
Gambar 6.14 Flowchart Aku Boleh Pulang Ga, Dok?.....	42
Gambar 6.15 Flowchart Daftar Check-Up.....	44
Gambar 6.16 Flowchart Antrian Saya!.....	45
Gambar 6.17 Flowchart Minum Obat.....	45
Gambar 6.18 Flowchart Minum Penawar.....	46
Gambar 6.19 Flowchart Exit.....	46
Gambar 7.1 Inisialisasi Main Program.....	77
Gambar 7.2 Login sebagai Manager.....	77
Gambar 7.3 Login sebagai Dokter.....	78
Gambar 7.4 Login sebagai Pasien.....	78
Gambar 7.5 Tidak Ada Username Terdaftar.....	78
Gambar 7.6 Kredensial Salah.....	79
Gambar 7.7 Register Berhasil.....	79
Gambar 7.8 Register Gagal.....	79
Gambar 7.9 Logout.....	80
Gambar 7.10 Password Berhasil Diubah.....	80
Gambar 7.11 User Tidak Terdaftar.....	81
Gambar 7.12 Kode Unik Salah.....	81
Gambar 7.13 Help Belum Login.....	81
Gambar 7.14 Help Manager.....	82
Gambar 7.15 Help Dokter.....	82
Gambar 7.16 Help Pasien.....	82
Gambar 7.17 Lihat Denah.....	83

Gambar 7.18 Lihat Ruangan.....	83
Gambar 7.21 Lihat Antrian.....	86
Gambar 7.22 Berhasil Tambah Dokter.....	86
Gambar 7.23 Tidak Berhasil Tambah Dokter.....	87
Gambar 7.24 Berhasil Assign Dokter ke Ruangan.....	87
Gambar 7.25 Dokter Sudah Diassign ke Ruangan Lain.....	87
Gambar 7.26 Ruangan Sudah Ditempati.....	87
Gambar 7.27 Ruangan Sudah Ditempati dan Dokter Sudah Diassign.....	88
Gambar 7.28 Berhasil Diagnosis Pasien.....	88
Gambar 7.29 Pasien Tidak Terdiagnosis.....	89
Gambar 7.30 Pasien Berhasil Diobatin.....	90
Gambar 7.31 Pasien Tidak Berhasil Diobatin.....	90
Gambar 7.32 Pasien Berhasil Pulang.....	91
Gambar 7.33 Pasien Tidak Berhasil Pulang.....	91
Gambar 7.34 Pasien Berhasil Daftar Check Up.....	92
Gambar 7.35 Pasien Sudah Pernah Daftar Check Up.....	92
Gambar 7.36 Antrian Saya! (Ruangan penuh, antrian di luar ruangan).....	92
Gambar 7.37 Antrian Saya! (Belum melakukan DAFTAR_CHECKUP).....	93
Gambar 7.38 Antrian Saya! (Sedang berada di ruangan dokter).....	93
Gambar 7.39 Minum Obat (Pilihan Invalid).....	93
Gambar 7.40 Minum Obat (Inventaris Pasien Isi).....	94
Gambar 7.41 Minum Obat (Inventaris Pasien Kosong).....	94
Gambar 7.42 Minum Penawar (Stack IsiPerut Isi).....	94
Gambar 7.43 Minum Penawar (Stack IsiPerut Kosong).....	95
Gambar 7.44 Exit.....	95
Gambar 7.45 Aura.....	96

DESKRIPSI PERSOALAN

Tugas besar ini meminta kami untuk membuat sebuah program sistem manajemen rumah sakit khusus yang dirancang untuk menangani pasien unik, yaitu para Nimon—makhluk kecil kuning yang sering mengalami kecelakaan karena tingkah mereka sendiri. Sistem ini bertujuan untuk membantu Dr. Neroifa dalam mengelola rumah sakit baru yang ia bangun bersama Gro dan Luiy, setelah menyadari bahwa para Nimon sering terluka dan belum ada sistem medis yang tertata dengan baik.

Dalam program ini, akan terdapat berbagai fungsionalitas seperti proses login dan registrasi pasien, melihat denah rumah sakit, menambah dokter, mendiagnosis penyakit, memberikan obat, serta mengelola antrian pemeriksaan pasien. Pasien dapat mendaftar check-up, mengonsumsi obat, meminum penawar jika salah obat, dan berkonsultasi untuk pulang. Di sisi lain, dokter dapat menangani diagnosis dan pengobatan, sementara manajer dapat mengelola data pengguna dan melihat semua antrian. Kami diminta menggunakan berbagai struktur data seperti Set, Map, Stack, Queue, dan List untuk menyimpan dan memproses data secara efisien.

Dengan menyelesaikan program ini, kami diharapkan mampu mengorganisasi kekacauan yang ada menjadi sistem rumah sakit digital yang rapi, fungsional, dan dapat membantu semua pengguna—baik pasien, dokter, maupun manajer—dalam aktivitas medis di dunia para Nimon.

RENCANA IMPLEMENTASI

Tabel 1.1 Rencana Implementasi

Implementasi ADT	FITUR	Deskripsi Implementasi	Alasan Implementasi
ADT Set, ADT List Statis	F01 - Login	<p>UserSet menyimpan seluruh data user dalam bentuk array, serta memastikan tidak ada duplikasi username.</p> <p>UserSet menggunakan array statis ‘User users[MAX_USER S]’ sebagai penyimpanan data user, sehingga pengaksesan cepat dan terstruktur.</p>	<p>Penggunaan ADT Set dan List Statis memudahkan validasi data login dengan cepat karena pencarian dilakukan dalam memori. Selain itu, struktur ini sesuai untuk sistem dengan jumlah user terbatas dan memperjelas batasan sistem (maksimal MAX_USERS). Hal ini juga mendukung modularitas dan pengujian program tanpa tergantung file eksternal.</p>
ADT Set, ADT List Statis	F02 - Register	<p>UserSet menyimpan data user baru yang berisikan username password dan role ke dalam array statis ‘User users[MAX_USER S]’</p> <p>Saat melakukan registrasi, sistem</p>	<p>Penggunaan ADT Set (untuk memastikan keunikan username) dan List Statis (untuk penyimpanan data user dalam array) memudahkan manajemen user tanpa perlu alokasi dinamis atau file</p>

		<p>akan memeriksa apakah username sudah ada atau belum menggunakan linear search dan menambahkan user baru ke array jika tidak ada. Fitur ini memastikan tidak ada duplikasi username dan jumlah user tidak melebihi batas maksimum</p>	<p>eksternal. Hal ini sesuai untuk sistem dengan jumlah user terbatas dan membuat pengujian lebih mudah dan cepat.</p>
ADT User Sederhana	F03 - Logout	<p>mengecek 'current_user' apakah memiliki isi, dan kemudian mengkosongkannya lagi menjadi NULL.</p>	<p>Logout cukup dengan mengatur ulang pointer current_user tanpa perlu menghapus atau mengubah data user, sehingga implementasi tetap efisien dan sederhana. ADT User menyimpan semua informasi login, sehingga pointer ke objek ini cukup untuk menjaga konteks pengguna selama sesi berlangsung.</p>
ADT List Statik	F04 - Lupa Password	<p>Digunakan untuk mencari user di dalam 'UserSet' menggunakan fungsi 'findUser'.</p> <p>Digunakan untuk membandingkan kode unik hasil inputan user dengan rleCode hasil</p>	<p>Karena jumlah maksimum untuk user (MAX_USER) dan panjang passwordnya (MAX_LEN) sudah di-define, sehingga tidak memerlukan array dinamis. Pencarian user dengan data set</p>

		'generateRLE'	yang relatif kecil ini lebih cepat dilakukan dengan pencarian linear melalui indeks. Penggunaan ADT List Statik ini juga sudah memenuhi kriteria dan lebih aman untuk dipakai karena tidak ada alokasi memori.
ADT User Sederhana	F05 - Menu & Help	Mengecek apakah current_user menunjuk pada user yang login. Jika tidak (NULL), maka tampilkan menu bantuan dasar. Jika ya, tampilkan menu sesuai current_user->role.	ADT User menyimpan informasi peran (role) sehingga cukup dengan memeriksa current_user untuk menentukan hak akses dan jenis bantuan yang relevan.
ADT List Statis, ADT Map, dan ADT Set	F06 - Denah Rumah Sakit	Digunakan untuk mengakses jumlahRuangan dari fitur LIHAT_DENAH Digunakan ADT List, Map, dan Set dalam fitur LIHAT_RUANGA N	Menggunakan ADT yang dapat merepresentasikan kumpulan ruangan dalam denah Menggunakan ADT list untuk validasi inputan id Ruangan dan mengambil kapasitas dan informasi pasien dari elemen ruangan. Selain itu, digunakan ADT Map dalam pembuatan logika <i>pairing</i> antara idDokter dan

			idRuangan. Mengambil detail username dokter dan pasien menggunakan ADT Set.
ADT List Statis	F07 - Lihat User	<p>UserList menggunakan array statis User users[MAX_USER S] untuk menyimpan semua data user, termasuk ID, Nama, Role, dan Penyakit.</p> <p>Fitur ini menampilkan semua user dalam bentuk tabel, dapat diurutkan berdasarkan ID atau Nama secara ascending atau descending untuk memudahkan pencarian dan monitoring data.</p>	<p>Penggunaan ADT List Statis memudahkan pengaksesan data user karena array statis memungkinkan iterasi cepat. Selain itu, array statis lebih sederhana untuk sistem dengan jumlah user terbatas dan terdefinisi (maksimal MAX_USERS).</p> <p>Struktur ini juga mendukung modularitas kode tanpa ketergantungan file eksternal, sesuai dengan desain sistem yang sederhana dan cepat.</p>
ADT List Statis	F08 - Cari User	<p>UserList menyimpan data user dalam array statis User users[MAX_USER S].</p> <p>Untuk pencarian user berdasarkan ID, digunakan algoritma binary search karena data</p>	<p>Implementasi binary search pada ID mempercepat pencarian karena array sudah diurutkan. Untuk pencarian Nama, sequential search memudahkan pencarian pada data nama yang mungkin belum</p>

		<p>user diurutkan berdasarkan ID.</p> <p>Untuk pencarian user berdasarkan Nama, digunakan sequential search agar dapat menangani data nama yang tidak berurutan.</p>	<p>terurut, dan cocok untuk sistem dengan jumlah user yang terbatas. Struktur ini mendukung modularitas, validasi cepat di memori, dan mempermudah implementasi fitur tanpa tergantung file eksternal.</p>
<p>ADT Denah, ADT Map (MapDokterRuangan), ADT Queue (Antrian Pasien), ADT Set (UserSet)</p>	F09 - Lihat Antrian	<p>Mengakses data dari struktur Denah (kumpulan ruangan), MapDokter (memetakan ID ruangan ke ID dokter), Queue (antrian pasien dalam ruangan), dan Set pengguna (UserSet). Setiap ruangan dicek apakah memiliki dokter atau antrian pasien, lalu ditampilkan informasi dokter dan daftar pasien berdasarkan posisi dalam antrian</p>	<p>ADT Denah memungkinkan pengelompokan ruangan secara sistematis. MapDokter memudahkan asosiasi dokter dengan ruangan tertentu. Queue digunakan karena sifat antrian pasien bersifat FIFO (First In First Out). Set digunakan untuk penyimpanan user yang memastikan tidak ada duplikasi dan memungkinkan pencarian user secara cepat berdasarkan ID. Penggunaan ADT-ADT ini mendukung modularitas dan efisiensi sistem, serta memudahkan validasi dan pemrosesan data di memori.</p>

ADT Set, ADT Map, ADT List Statis	F10 - Tambah Dokter	<p>Menerima input untuk dokter baru dan menggunakan operasi ADT Set dalam register pada command TAMBAH_DOKTER</p> <p>Digunakan ADT Set, List, dan MAP pada command ASSIGN DOKTER</p>	<p>Karena cocok untuk mengelola <i>unique pairing</i> dari informasi pengguna.</p> <p>Tugas utama dari fitur ini yaitu membuat pemetaan untuk assign dokter ke ruangan dan didukung ADT Set dan List sebagai validator.</p>
ADT List Statis, ADT Queue	F11 - Diagnosis	<p>Fungsi diagnosisPasien memeriksa pasien pertama di antrian ruangan dokter.</p> <p>- Data pasien dalam DataCheckUp dibandingkan dengan semua entri DataPenyakit</p> <p>Jika seluruh parameter cocok, pasien didiagnosis menderita penyakit tertentu.</p> <p>- Hasil diagnosis disimpan dalam idPenyakit pasien dan dicetak di layar.</p>	<p>Menggunakan ADT List Statis dalam DataPenyakit karena jumlah penyakit tetap dan terbatas, memungkinkan pencarian cepat.</p> <p>Queue pada antrianPasien memastikan pasien didiagnosis berdasarkan urutan kedatangan.</p> <p>Pencocokan parameter berbasis range efektif untuk diagnosis berbasis data medis.</p>
ADT Queue, ADT Map	F12 - Ngobatin	<p>Fungsi berobatPasien memproses pasien pertama dari QUEUE yang sebelumnya telah didiagnosis.</p> <p>Penyakit pasien digunakan untuk mencari obat melalui</p>	<p>ADT Map dalam MapPenyakitObat mempermudah pemetaan penyakit ke obat dan urutan resep.</p> <p>ADT List Statis pada DataObat dipilih karena jumlah obat terbatas dan pencarian</p>

		<p>MapPenyakitObat berdasarkan ID penyakit dan urutan pemberian obat. Nama obat diambil dari DataObat dan ditampilkan secara terurut. Setelah itu, pasien dikeluarkan dari antrian (dequeue) dan proses dianggap selesai.</p>	<p>berdasarkan ID dapat dilakukan cepat. Queue memastikan bahwa proses berobat mengikuti urutan antrian yang valid. Memastikan validasi diagnosis dilakukan sebelum pengobatan.</p>
ADT Queue, ADT List Statis, ADT Stack, ADT Map	F13 - Aku boleh pulang ga, dok?	<p>Fitur mencari entri pasien dari antrian (Queue) berdasarkan ID menggunakan fungsi getEntriPasien(). Obat dalam perut pasien diperiksa melalui Stack lalu dicocokkan dengan resep dari MapPenyakitObat. Jika semua obat sesuai urutan, pasien diizinkan pulang dan dikeluarkan dari antrean.</p>	<p>Queue digunakan untuk menyimpan antrean pasien berdasarkan FIFO. Stack merepresentasikan urutan konsumsi obat (LIFO). Map digunakan untuk mencocokkan penyakit dan resep. List statis memudahkan iterasi pasien dan penyakit yang jumlahnya tetap dan terbatas.</p>
ADT List Statik, ADT Map, ADT Set, ADT Queue	F14 - Daftar Check-Up	<p>Fungsi isPasienTerdaftar menggunakan ADT Queue untuk mengecek posisi ID pasien di antrianPasien dalam denah getRuanganByDokter menerapkan ADT Map untuk mendapatkan</p>	<p>ADT Queue cocok dipakai karena antrian pasien bersifat FIFO. Queue cocok untuk dipakai dalam penyimpanan data pasien yang mengantri berdasarkan urutan daftar check-up ADT Map</p>

		<p>ruangan dari ID dokter (mencocokkan antar ruangan dan dokter) ADT Set disini dipakai dalam function findUserById untuk mengambil informasi user dari ID</p>	<p>merupakan tipe ADT yang paling cocok untuk menghubungkan dan mencocokkan ID dokter dengan ruangnya, mempercepat pencarian juga ADT Set memastikan semua data user tersimpan (pasien dan dokter) dan memungkinkan akses melalui ID ADT List Statis digunakan untuk menyimpan daftar ruangan dalam denah, memudahkan iterasi untuk mencari pasien atau dokter</p>
<p>ADT List Statik, ADT Map, ADT Set, ADT Queue</p>	<p>F15 - Antrian Saya</p>	<p>Pada prosedur ini, ADT List digunakan untuk iterasi dan validasi ruangan, ADT Queue untuk pengecekan status dan posisi pasien dalam antrian, ADT Map untuk mendapatkan data dokter sesuai dengan ruangan, dan ADT Set untuk mengambil data username dan role user yang sedang login</p>	<p>ADT List Statis digunakan untuk mengakses daftar ruangan dan antrian pasien di tiap ruangan pada denah rumah sakit</p> <p>ADT Queue digunakan untuk membuat antrian pasien di setiap ruangan dengan konsep FIFO, sehingga dapat mengetahui posisi pasien dalam antrian</p> <p>ADT Map</p>

			<p>digunakan untuk menghubungkan idRuangan dengan idDokter yang bertugas, memudahkan pengambilan dokter berdasarkan ruangan</p> <p>ADT Set digunakan untuk mengelola user, yang memungkinkan pencarian user berdasarkan ID</p>
ADT List Statik, ADT Map, ADT User, ADT UserSet	F16 - Minum Obat	<p>ADT List Statis digunakan sebagai struktur data Inventaris Obat yang dimiliki oleh pasien. Fitur ini akan menampilkan daftar obat dari list ini dan menghapus elemen yang dipilih oleh pasien. ADT Stack diimplementasikan untuk merepresentasikan "IsiPerut" Pasien. Obat yang diminum akan di-push ke dalam stack ini, mengikuti prinsip Last-In, First-Out (LIFO). ADT User Sederhana digunakan untuk mengakses data current_user yang sedang login, khususnya pada</p>	<p>ADT List Statis cocok untuk merepresentasikan inventaris karena secara alami menggambarkan sebuah koleksi. Penggunaan list statis (berbasis array) konsisten dengan implementasi fitur lain dalam proyek, mudah dikelola, dan efisien untuk jumlah obat per pasien yang terbatas. ADT Stack wajib digunakan untuk "perut" karena fitur F17 (Minum Penawar) harus dapat mengambil kembali obat yang terakhir diminum. Sifat LIFO dari Stack memastikan</p>

		field inventaris dan "IsiPerut"nya.	operasi ini berjalan sesuai spesifikasi.
ADT List Statik, ADT User, ADT Set	F17 - Minum Penawar	ADT Stack digunakan untuk mengakses "Perut" Pasien. Fitur ini akan melakukan operasi pop untuk mengambil obat teratas (yang terakhir diminum) dari stack. ADT List Statis digunakan sebagai tujuan pengembalian obat. Obat yang telah di-pop dari "perut" akan di-insert kembali ke dalam Inventaris Obat pasien. ADT User Sederhana digunakan untuk mengakses data <code>current_user</code> untuk memodifikasi <code>perut_pasien</code> dan <code>inventory_obat</code> .	ADT Stack adalah pilihan yang paling logis dan efisien untuk fitur penawar. Kebutuhan untuk mengambil elemen terakhir yang dimasukkan adalah definisi dari operasi pop pada Stack, sehingga memastikan fungsionalitas inti fitur ini terpenuhi dengan benar. ADT List Statis digunakan kembali sebagai tempat penyimpanan obat yang dikembalikan. Hal ini menjaga konsistensi data, di mana semua obat yang dimiliki pasien selalu berada di dalam inventaris.

PEMBAGIAN KERJA

Tabel 2.1 Pembagian Kerja

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
F01 - Login	procedure loginUser(), ADT List Statik, ADT set	18224100 18224124	18224100 18224124	18222009 18224040 18224100 18224124
F02 - Register	procedure registerUser(), ADT List Statik, ADT set	18224124	18224124	18222009 18224040 18224100 18224124
F03 - Logout	procedure logoutUser(), ADT List	18224100	18224100	18222009 18224040 18224100 18224124
F04 - Lupa Password	procedure generateRLE(), function findUser(), procedure registerUser(), procedure lupaPassword() ADT List Statik	18224040	18224040	18222009 18224040 18224100 18224124
F05 - Menu & Help	procedure help(), ADT user sederhana	18222014	18222014	18222014 18221009
F06 - Denah Rumah Sakit	ADT List Statik, procedure initDenah(), function getPanjangDena h(), function getKapasitasDen ah(), procedure lihatDenah(), procedure lihatRuangan()	18221009	18221009	18221009
F07 - Lihat User	procedure lihatUser(), ADT List Statis, array	18224124	18224124, 18221009	18224124, 18221009

	users[MAX_USER S], pengurutan (ID>Nama)			
F08 - Cari User	procedure cariUser(), ADT List Statis, array users[MAX_USER S], binary search (ID), sequential search (Nama)	18224124	18224124, 18221009	18224124, 18221009
F09 - Lihat Antrian	procedure lihatSemuaAntria n()	18221009	18221009	18221009
F10 - Tambah Dokter	ADT List Statik, ADT Map, procedure initMapDokterRu angan(), function bool assignDokterToR uangan(), function bool unassignDokter() , function getRuanganByDo kter(), function getDokterByRuan gan(), function isDokterAssigne dtoRuangan(), function isRuanganAssign edToDokter(), procedure managerTambah Dokter(), procedure assignDokter()	18221009	18221009	18221009
F11 - Diagnosis	ADT List Statik, ADT Queue, procedure initPenyakit(), function getNamaPenyaki t(), procedure diagnosisPasienn()	18224072	18224072, 18221009	18224072, 18221009

)			
F12 - Ngobatin	ADT Queue, ADT Map, procedure initObat(), function getNamaObat(), procedure berobatPasien()	18224072	18224072, 18221009	18224072, 18221009
F13 - Aku boleh pulang ga, dok?	procedure F13_PulangDok(), function getEntriPasien(),	18222014	18222014, 18221009	18222014, 18221009
F14 - Daftar Check-Up	procedure clearInputBuffer(), function getPositiveFloat(), function getPositiveInt(), function getPositiveTekan andarah, function isPasienTerdaftar(), procedure daftarCheckUp()	18221009	18221009	18221009
F15 - Antrian Saya	ADT List Statik, ADT Map, ADT Set, ADT Queue, function isEmptyQueue(), function getPositionPasien(), function getSize(), function getDokterByRuang(), function findUserById()	18224040	18224040	18221009 18224040
F16 - Minum Obat	ADT List Statis (untuk inventaris), ADT Stack (untuk "perut"), procedure F16_MinumObat(), function getNamaObat(), function	18224100	18224100	18224100

	isListInventoryEmpty(), procedure deleteAtListInventory(), procedure pushStackPerut()			
F17 - Minum Penawar	ADT List Statis (untuk inventaris), ADT Stack (untuk "perut"), procedure F17_MinumPenawar(), function getNamaObat(), function isStackPerutEmpty(), procedure popStackPerut(), procedure insertLastListInventory()	18224100	18224100	18224100
F18 - Exit	-	18224072	18224072	18221009 18224072
B03 - Aura	add new struct ADT User, feat pada fitur F02, F12, F13, F07	18221009	18221009	18221009

HASIL Pengerjaan

Tabel 3.1 Hasil Pengerjaan

Fitur	Desain	Implementasi	Testing
F01 - Login	V	V	V
F02 - Register	V	V	V
F03 - Logout	V	V	V
F04 - Lupa Password	V	V	V
F05 - Menu & Help	V	V	V
F06 - Denah Rumah Sakit	V	V	V
F07 - Lihat User	V	V	V
F08 - Cari User	V	V	V
F09 - Lihat Antrian	V	V	V
F10 - Tambah Dokter	V	V	V
F11 - Diagnosis	V	V	V
F12 - Ngobatin	V	V	V
F13 - Aku boleh pulang ga, dok?	v	v	V
F14 - Daftar Check-Up	V	V	V
F15 - Antrian Saya	V	V	V
F16 - Minum Obat	V	V	V
F17 - Minum	V	V	V

Penawar			
F18 - Exit	V	V	V

Keterangan: V: sudah selesai dikerjakan, X: dikerjakan, tapi belum selesai, -: tidak dikerjakan sama sekali.

DESIGN COMMAND

1. F01 - Login

Command : Login

Masukan(I.S.): Data UserSet (user yang terdaftar) sudah ada, input username dan password.

Keluaran(F.S.): Apabila sesuai dengan yang terdaftar “Login Berhasil”, apabila tidak sesuai “Login Gagal”.

2. F02 - Register

Command : Register

Masukan : Mengisi Data UserSet (Username, Password, Role)

Keluaran : Apabila sudah ada yang terdaftar, “Akun sudah teregistrasi”. apabila tidak, “Akun berhasil di registrasi”

3. F03 - Logout

Command : Logout

Masukan(I.S.) : Sudah login sebelumnya. Meminta untuk logout.

Keluaran(F.S.) : Ter-Logout dari user tersebut.

4. F04 - Lupa Password

Command : Lupa Password

Masukan(I.S.) : Data UserSet berisi user yang terdaftar, proses pembaharuan password belum dijalankan.

Keluaran(F.S.) : Jika user ditemukan dan rleCode sesuai, password user diperbaharui ; Jika tidak sesuai, tidak ada perubahan dalam UserSet

5. F05 - Menu & Help

Command : Menu & Help

Masukan : Pengguna berada dalam kondisi login atau belum login.

Keluaran : Sistem menampilkan daftar perintah yang dapat dijalankan sesuai status login dan peran pengguna.

6. F06 - Denah Rumah Sakit

Command : Denah Rumah Sakit

Masukan : Denah yang di-generate ketika menjalankan main program, idRuangan yang ingin dilihat, set User yang di-generate ketika menjalankan main program, dan map dokter.

Keluaran : Sistem menampilkan denah yaitu banyak ruangan yang diinisialisasi sebelumnya serta detail tiap ruangan (kapasitas ruangan, dokter, dan pasien dalam ruangan)

7. F07 - Lihat User

Command : Lihat User

Masukan : Pilihan urutan data (ID atau Nama), Pilihan sort (ASC/DESC)

Keluaran : Tabel data user (ID, Nama, Role, Penyakit) sesuai pilihan urutan dan sort

8. F08 - Cari User

Command : Cari User

Masukan : Pilihan pencarian (ID atau Nama atau Penyakit), Data yang dicari (ID>Nama>Penyakit)

Keluaran : Data user (ID, Nama, Role, Penyakit) jika ditemukan dan Pesan "User tidak ditemukan" jika data tidak ditemukan

9. F09 - Lihat Antrian

Command : Lihat Semua Antrian

Masukan : denah, set, mapDokter

Keluaran : Sistem menampilkan keseluruhan ruangan dan isinya, meliputi dokter yang bertugas, pasien yang di dalam ruangan, serta pasien di antrian

10. F10 - Tambah Dokter

Command : Tambah Dokter

Masukan : idDokter dan idRuangan

Keluaran : Sistem melakukan validasi register untuk menambah dokter yang bertugas ke dalam program oleh manager. Lalu, sistem melakukan assign dokter ke ruangan.

11. F11 - Diagnosis

Command : Diagnosis

Masukan : idPasien dan idPenyakit

Keluaran : Sistem menampilkan diagnosis penyakit yang diderita pasien

12. F12 - Ngobatin

Command : Ngobatin
Masukan : idPasien, idPenyakit, daftarMapObat
Keluaran : Sistem menampilkan obat yang perlu dikonsumsi pasien

13. F13 - Aku boleh pulang ga, dok?

Command : Aku boleh pulang ga, dok?
Masukan : current_user sebagai pasien, data denah rumah sakit
Keluaran : Sistem mengecek apakah pasien sudah didiagnosis dan minum obat sesuai resep. Jika ya, pasien dikeluarkan dari antrian dan boleh pulang. Jika tidak, menampilkan alasannya.

14. F14 - Daftar Check-Up

Command : Daftar Check Up
Masukan : set, denah, mapDokter, currentUser sebagai parameter fungsi serta di dalam fungsi ada masukan entri data check up pasien
Keluaran : Sistem mampu meng-assign pasien yang telah melakukan inputan check up dengan valid ke dokter yang tersedia

15. F15 - Antrian Saya!

Command : Antrian Saya!
Masukan(I.S) : Data UserSet, Denah, MapDokterRuangan, dan User (User sebagai pasien)
Keluaran(F.S) : Sistem menampilkan status antrian pasien jika sedang mengantri, di ruangan, atau belum terdaftar

16. F16 - Minum Obat

Command : Minum Obat
Masukan : Pengguna telah login sebagai Pasien. Data inventory_obat (Inventaris Obat) milik pasien terdefinisi
Keluaran : Jika inventaris tidak kosong dan pilihan valid, obat yang dipilih akan pindah dari inventory obat ke “perut” pasien. Sistem menampilkan pesan keberhasilan dan menampilkan sisa obat yang ada di Inventory pasien.

17. F17 - Minum Penawar

Command : Minum Penawar

Masukan : Pengguna telah login sebagai Pasien. Data perut_pasien milik pasien terdefinisi.

Keluaran : Jika tidak kosong, sistem akan mengembalikan obat terakhir diminum ke Inventory pasien. Jika kosong akan menampilkan pesan tidak ada yang dimuntahkan.

18. F18 - Exit

Command : Exit

Masukan : -

Keluaran : Pesan berhasil keluar atau Kembali ke menu awal

DESIGN KAMUS DATA

1. F01 - Login

```
type User : <id : int
            username : string,
            password : string,
            role : enum {MANAGER, DOKTER, PASIEN}>

type UserSet : < users : array [0..MAX_USERS] of User,
                jumlahUser : integer >
current_user : pointer to User
```

2. F02 - Register

```
username, password, role, user_baru, access : string
data_user : array of strings
index, jumlahUser : integer
username_registered : boolean
```

3. F03 - Logout

```
current_user : pointer to User
```

4. F04 - Lupa Password

```
username, rleCode, newPassword, correctRle : string
user : pointer to User
```

5. F05 - Menu & Help

```
current_user : pointer to User
username : string
role : enum {MANAGER, DOKTER, PASIEN}
```

6. F06 - Denah Rumah Sakit

```
type Ruang : < idDokter : integer,
                idPasien array of [0..MAX_PASIEN] : integer,
                jumlahPasien : integer,
                idRuang : integer,
                antrianPasien : Queue >
```

```
type Denah : < ruangan array of [0..MAX_RUANGAN] : Ruangan,  
                jumlahRuangan : integer,  
                kapasitas : integer >
```

7. F07 - Lihat User

```
type User : <  
    id : int,  
    username : string,  
    password : string,  
    role : enum {MANAGER, DOKTER, PASIEN},  
    penyakit : string >  
  
type UserSet : <  
    users : array [0..MAX_USERS] of User,  
    jumlahUser : integer >
```

8. F08 - Cari User

```
type User : <  
    id : int,  
    username : string,  
    password : string,  
    role : enum {MANAGER, DOKTER, PASIEN},  
    penyakit : string >  
  
type UserSet : <  
    users : array [0..MAX_USERS] of User,  
    jumlahUser : integer >  
  
type SearchResult : <  
    id : int,  
    username : string,  
    role : enum {MANAGER, DOKTER, PASIEN},  
    penyakit : string >
```

9. F09 - Lihat Antrian

```
denah : const Denah*  
set : const UserSet*  
mapDokter : const MapDokterRuangan*
```

10. F10 - Tambah Dokter

```
type DokterRuangan : < idDokter : integer,  
                        idRuangan : integer >
```

```
type MapDokterRuangan : < map array of DokterRuangan  
[0..MAX_DOKTER],  
                jumlahDokter : integer >
```

11. F11 - Diagnosis

```
type DataPenyakit : < id : integer  
    array of char namaPenyakit  
[0..MAX_LEN_PENYAKIT]  
    suhu_min : float  
    suhu_max : float  
    sistol_min, sistol_max : integer  
    diastol_min, diastol_max : integer  
    detak_min, detak_max : integer  
    saturasi_min, saturasi_max : float  
    gula_min, gula_max : integer  
    berat_min, berat_max : float  
    tinggi_min, tinggi_max : integer  
    kolesterol_min, kolesterol_max : integer  
    trombosit_min, trombosi_max : integer >
```

12. F12 - Ngobatin

```
type DataObat : < idObat : integer  
    array of char namaObat [0..MAX_NAMA_OBAT] >  
  
type MapPenyakitObat : < idPenyakit : integer  
    idObat : integer  
    urutan : integer >
```

13. F13 - Aku boleh pulang ga, dok?

```
current_user : pointer to User  
  
type User : <  
    id : integer,  
    ...  
    penyakit : string,  
    perut_pasien : StackPerut,  
    inventory_obat : InventoryObat >  
  
type Denah : <  
    ruangan : array [0..MAX_RUANGAN-1] of Ruangan,  
    jumlahRuangan : integer,  
    kapasitas : integer >  
  
type Ruangan : <  
    antrianPasien : Queue >
```



```
type StackPerut : <
  data : array of integer,
  top : integer >
```

```
type EntriAntrianCheckup : <
  idPasien : integer,
  idPenyakit : integer,
  dataCheckUp : ... >
```

```
type MapPenyakitObat : <
  idPenyakit : integer,
  idObat : integer,
  urutan : integer >
```

14. F14 - Daftar Check-Up

```
type DataCheckUp : < idPasien : integer,
  suhuTubuh : float,
  tekananDarahSistol : integer,
  tekananDarahDiastol : integer,
  detakJantung : integer,
  saturasiOksigen : float,
  kadarGulaDarah : integer,
  beratBadan : float,
  tinggiBadan : integer,
  kadarKolesterolTotal : integer,
  kadarKolesterolLDL : integer,
  trombosit : integer >
```

```
type EntriAntrianCheckup : < idPasien : integer,
  nomorAntri : integer,
  dataCheckUp : DataCheckUp,
  idPenyakit : integer >
```

15. F15 - Antrian Saya!

```
set: pointer to UserSet
denah: pointer to Denah
mapDokter: pointer to MapDokterRuangan
currentUser: pointer to User
```

16. F16 - Minum Obat

```
type InventoryObat : <
  data: array[0..MAX_OBAT-1] of integer,
  jumlah: integer>
```

```
type StackPerut : <
```

```
data: array[0..MAX_OBAT-1] of integer,  
top: integer>  
type User : <  
id : int,  
username : string,  
password : string,  
role : enum {MANAGER, DOKTER, PASIEN},  
penyakit : string  
inventory_obat: InventoryObat,  
perut_pasien: StackPerut >
```

17. F17 - Minum Penawar

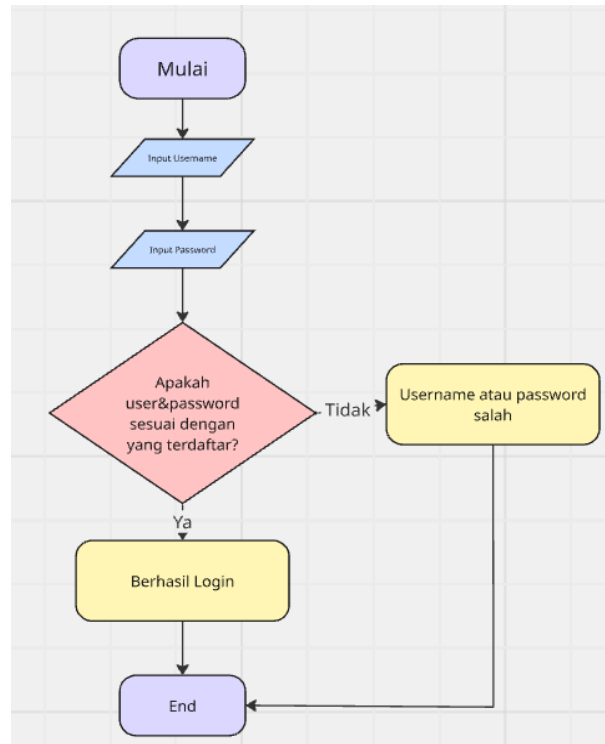
```
type InventoryObat : <  
data: array[0..MAX_OBAT-1] of integer,  
jumlah: integer>  
  
type StackPerut : <  
data: array[0..MAX_OBAT-1] of integer,  
top: integer>  
  
type User : <  
id : int,  
username : string,  
password : string,  
role : enum {MANAGER, DOKTER, PASIEN},  
penyakit : string  
inventory_obat: InventoryObat,  
perut_pasien: StackPerut >
```

18. F18 - Exit

```
input : char
```

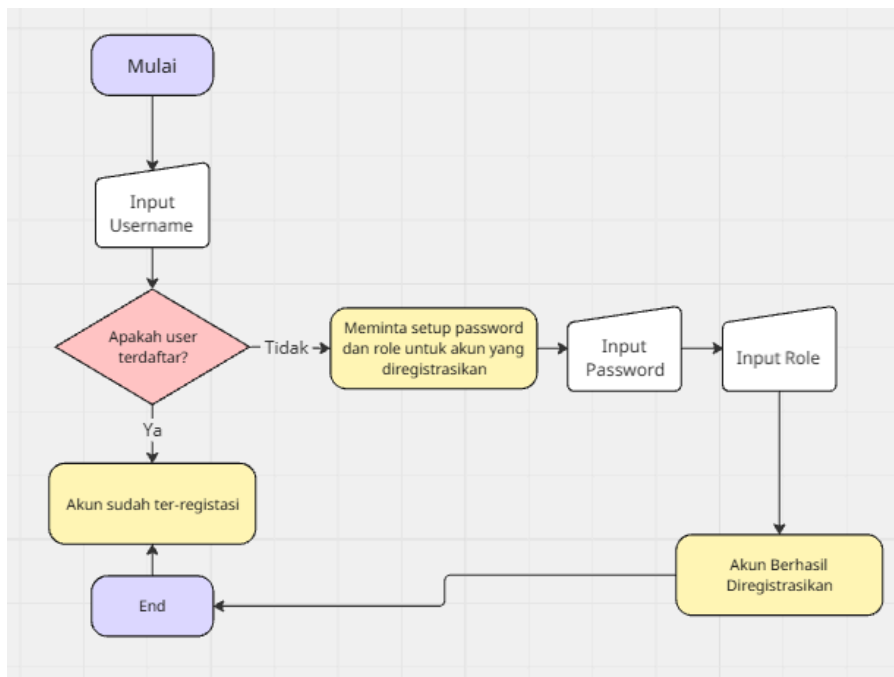
DESIGN DEKOMPOSISI

1. F01 - Login



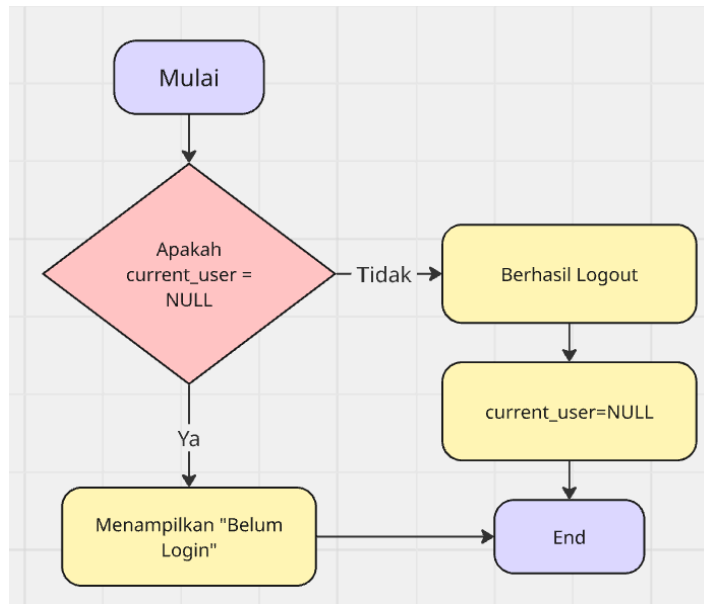
Gambar 6.1 Flowchart Login

2. F02 - Register



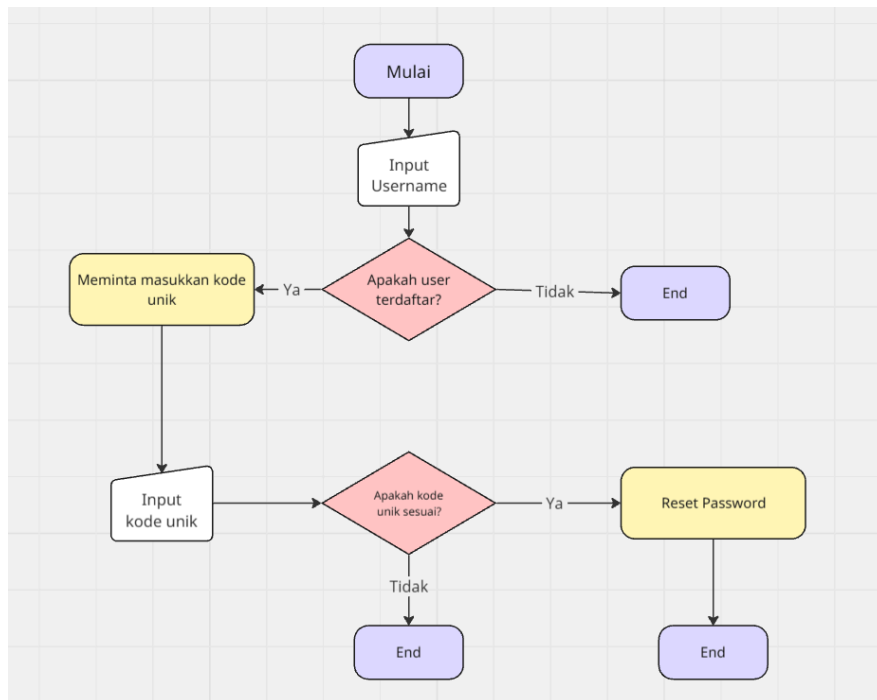
Gambar 6.2 Flowchart Register

3. F03 - Logout



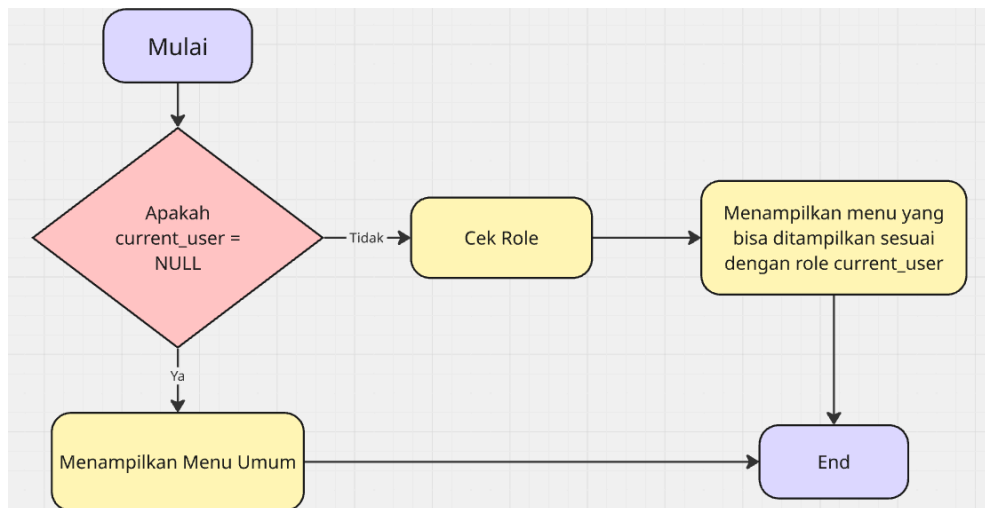
Gambar 6.3 Flowchart Logout

4. F04 - Lupa Password



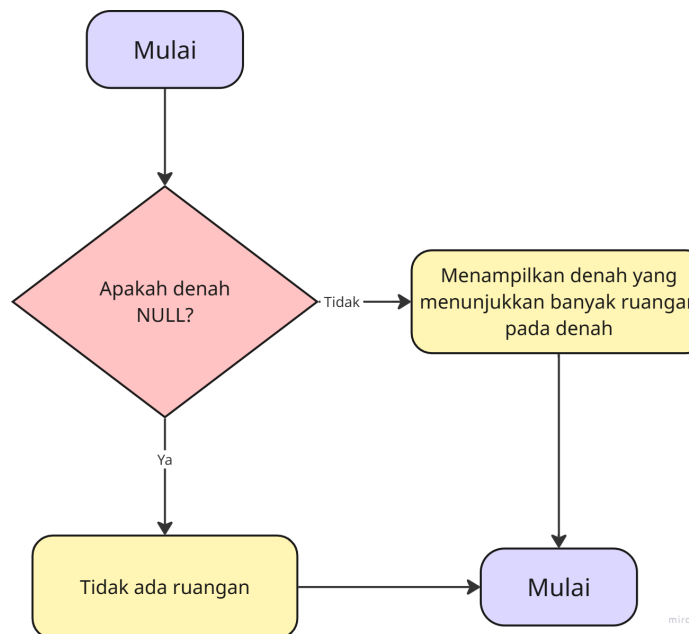
Gambar 6.4 Flowchart Lupa Password

5. F05 - Menu & Help

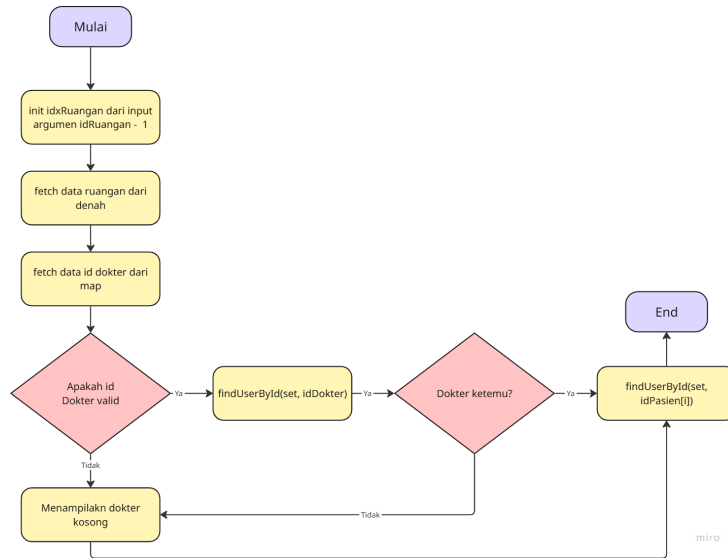


Gambar 6.5 Flowchart Menu & Help

6. F06 - Denah Rumah Sakit

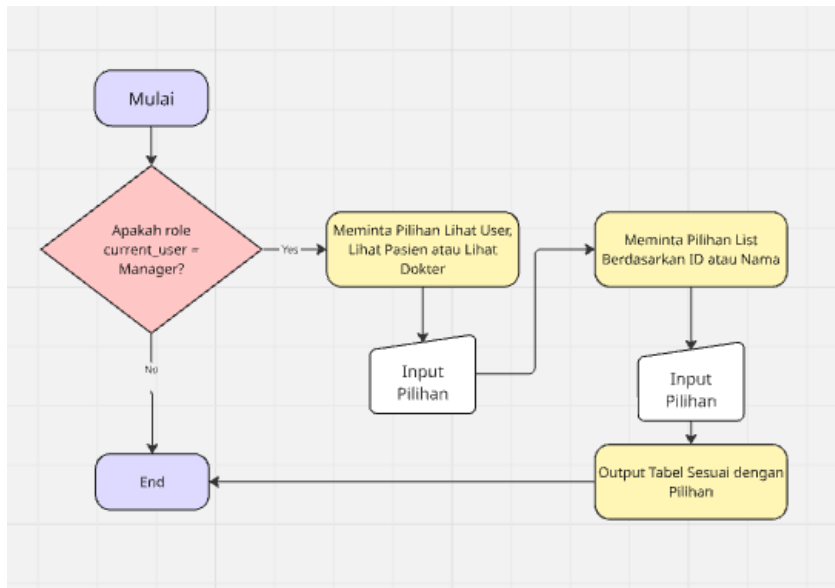


Gambar 6.6 Flowchart Lihat Denah



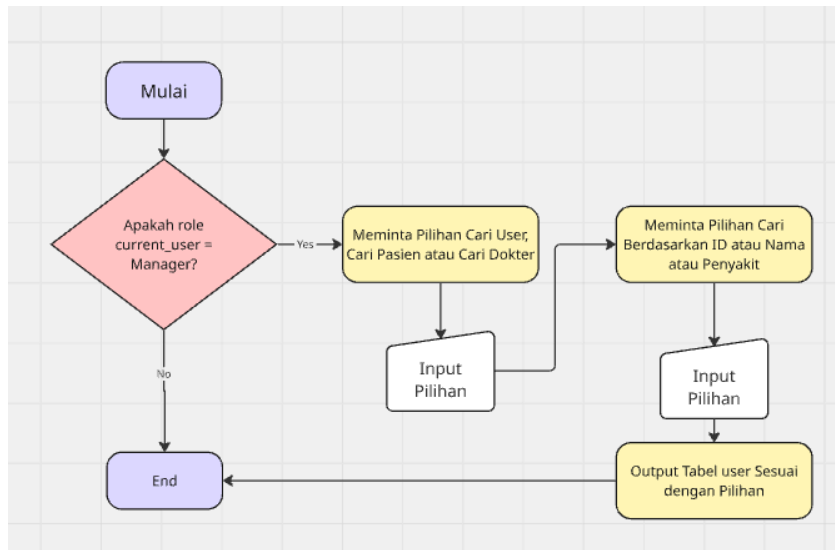
Gambar 6.7 Flowchart Lihat Ruangan

7. F07 - Lihat User



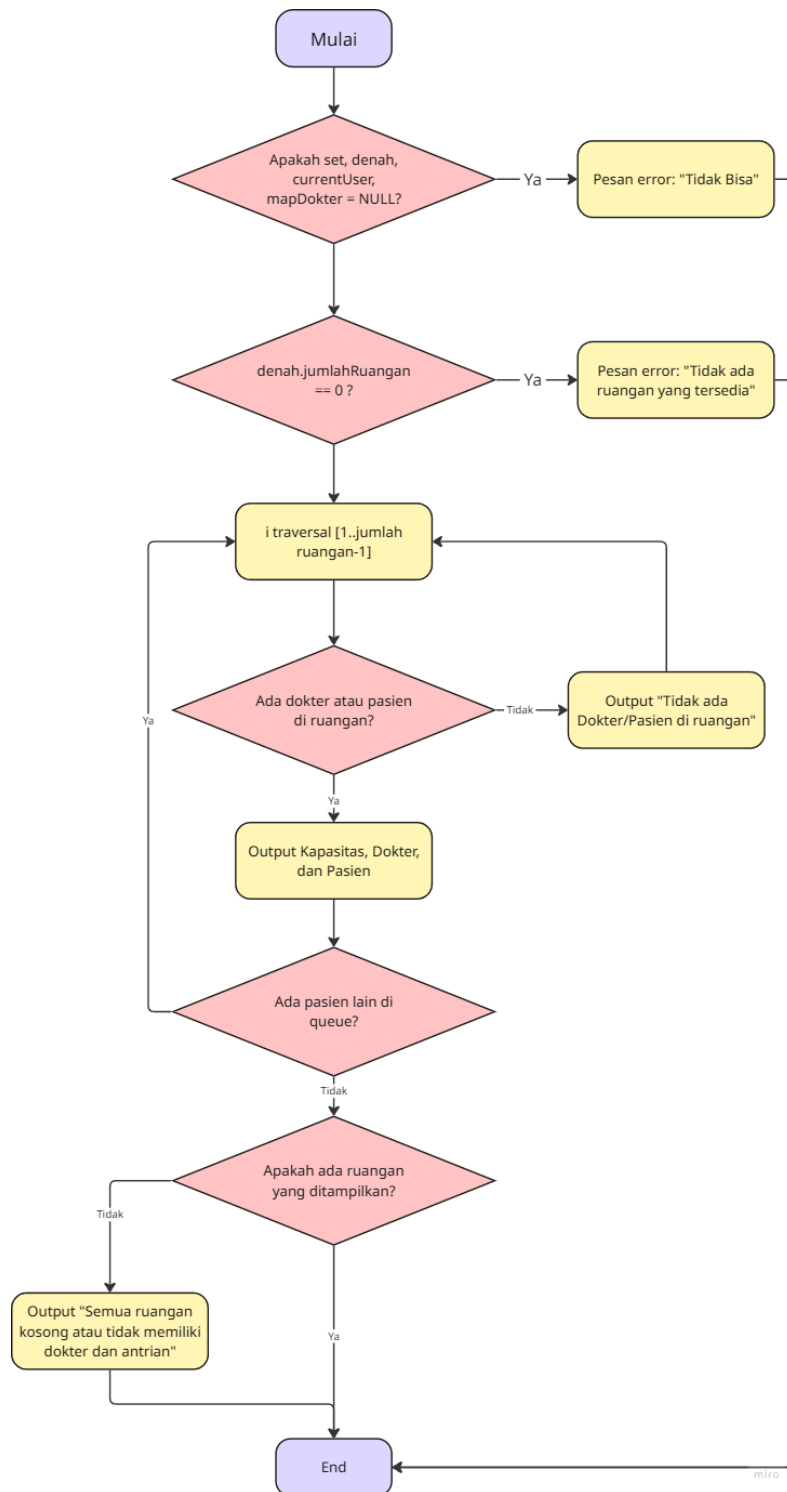
Gambar 6.8 Flowchart Lihat User

8. F08 - Cari User



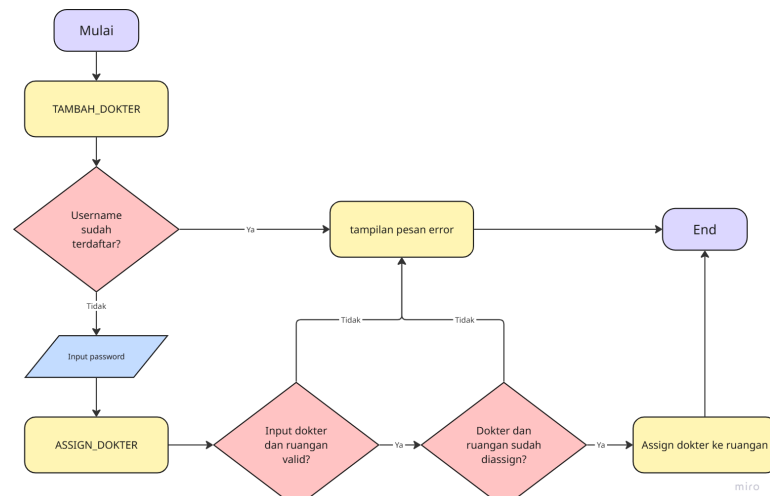
Gambar 6.9 Flowchart Cari User

9. F09 - Lihat Antrian



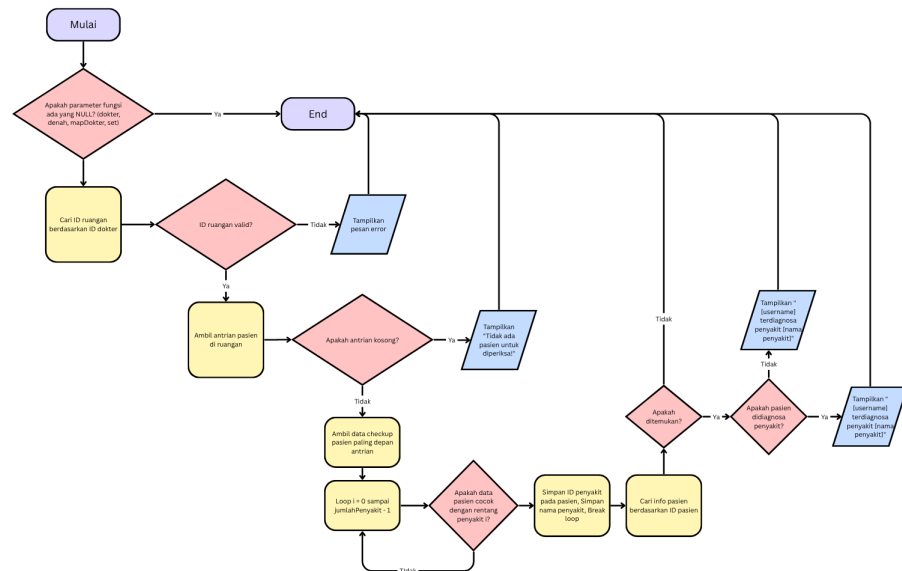
Gambar 6.10 Flowchart Lihat Antrian

10. F10 - Tambah Dokter



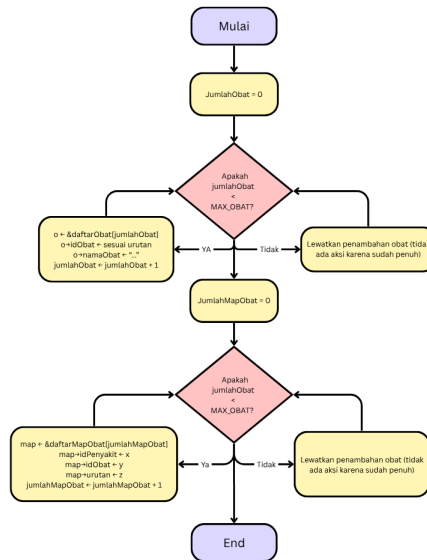
Gambar 6.11 Flowchart Tambah Dokter

11. F11 - Diagnosis



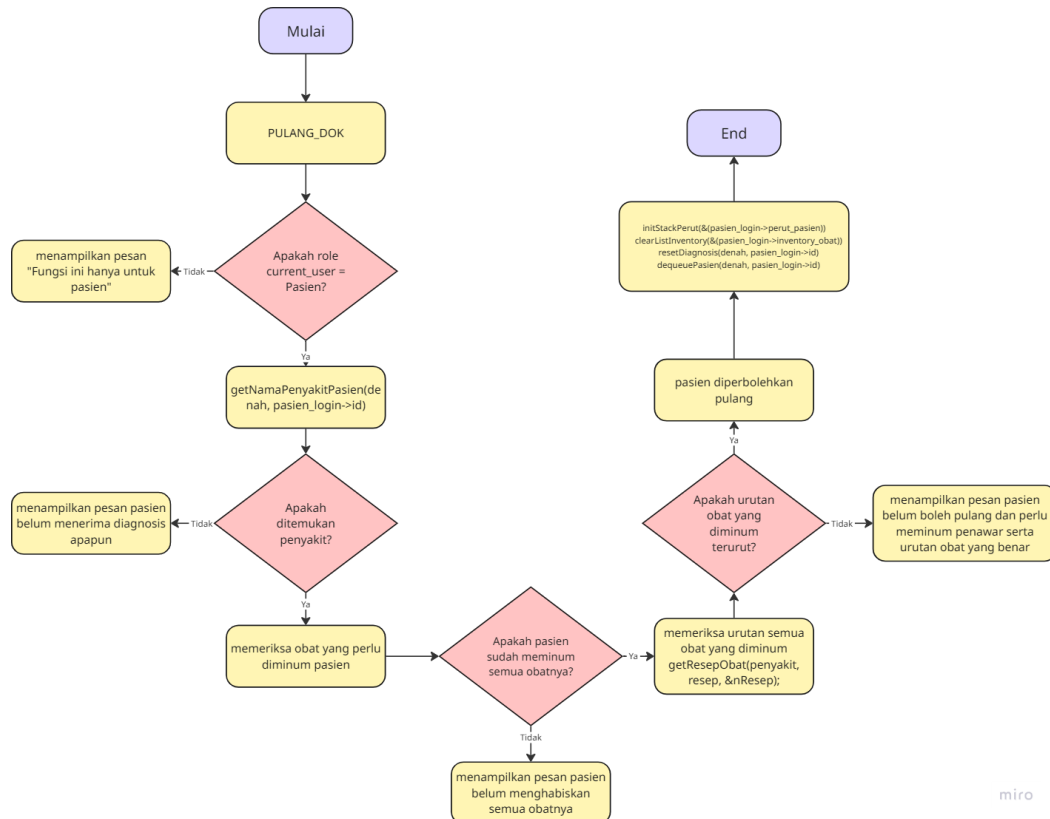
Gambar 6.12 Flowchart Diagnosis

12. F12 - Ngobatin



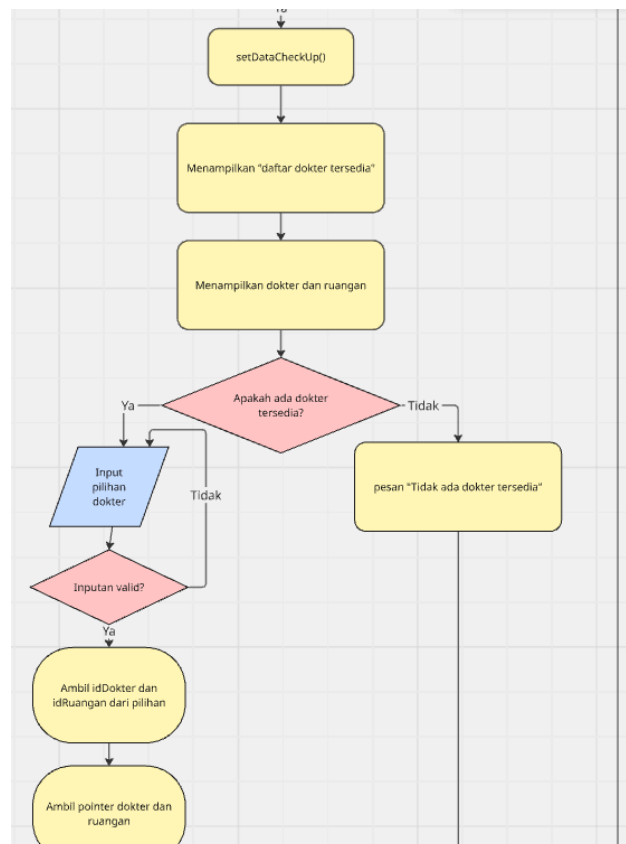
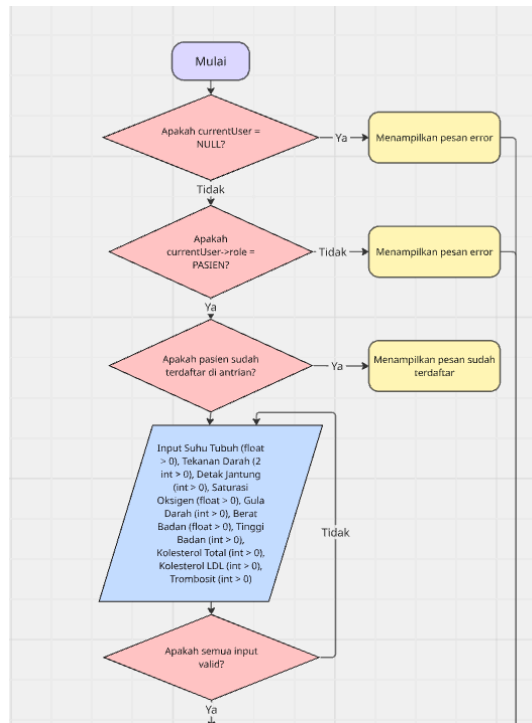
Gambar 6.13 Flowchart Ngobatin

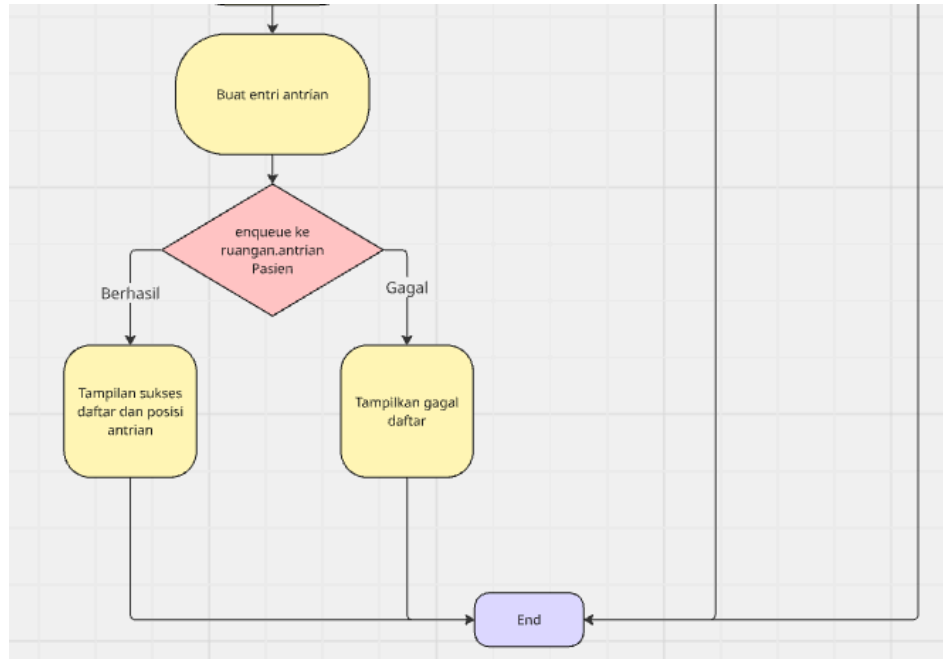
13. F13 - Aku boleh pulang ga, dok?



Gambar 6.14 Flowchart Aku Boleh Pulang Ga, Dok?

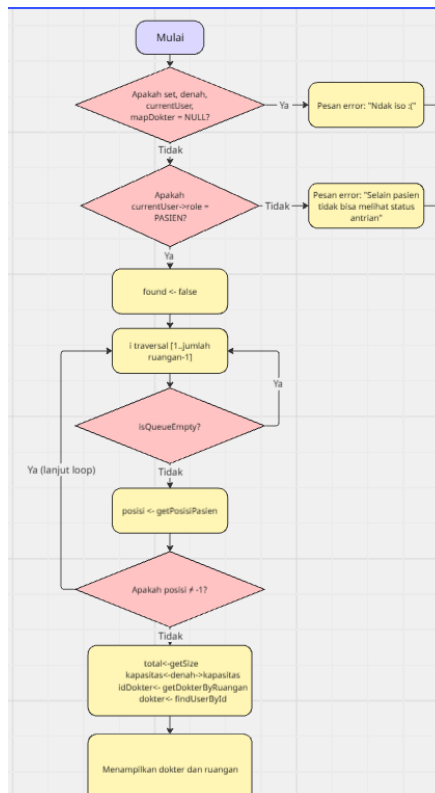
14. F14 - Daftar Check-Up

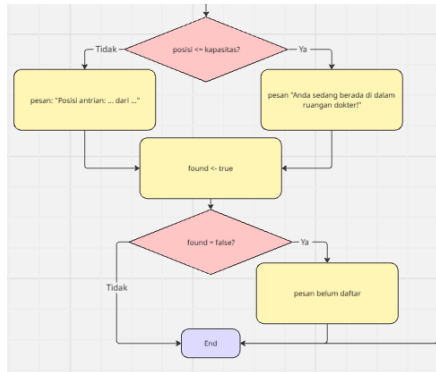




Gambar 6.15 Flowchart Daftar Check-Up

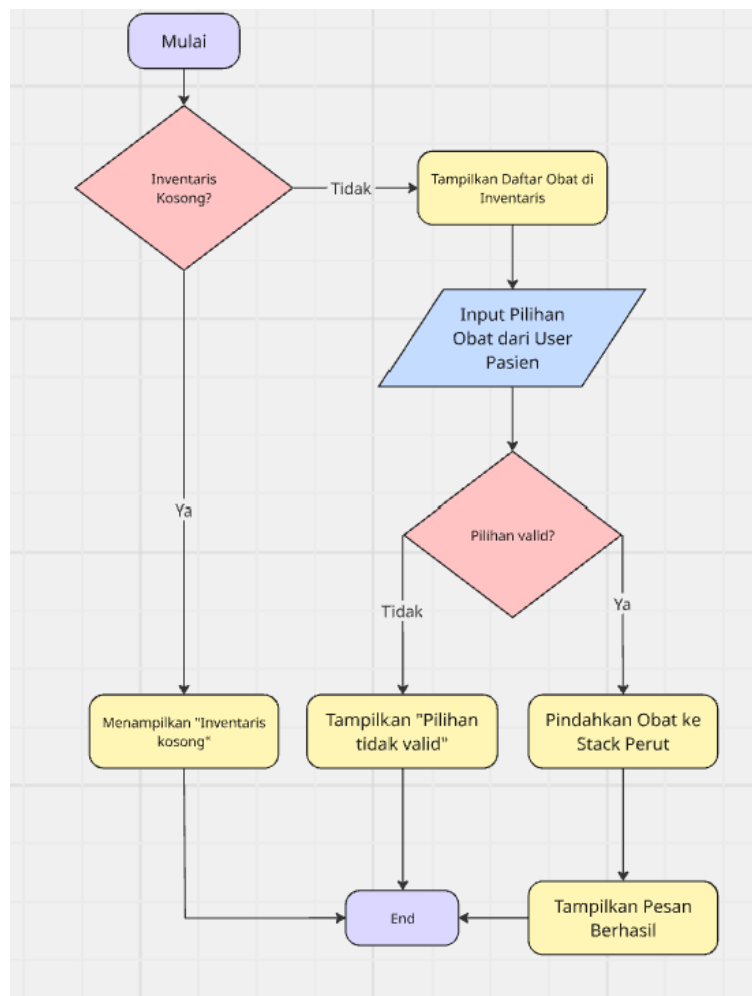
15. F15 - Antrian Saya!





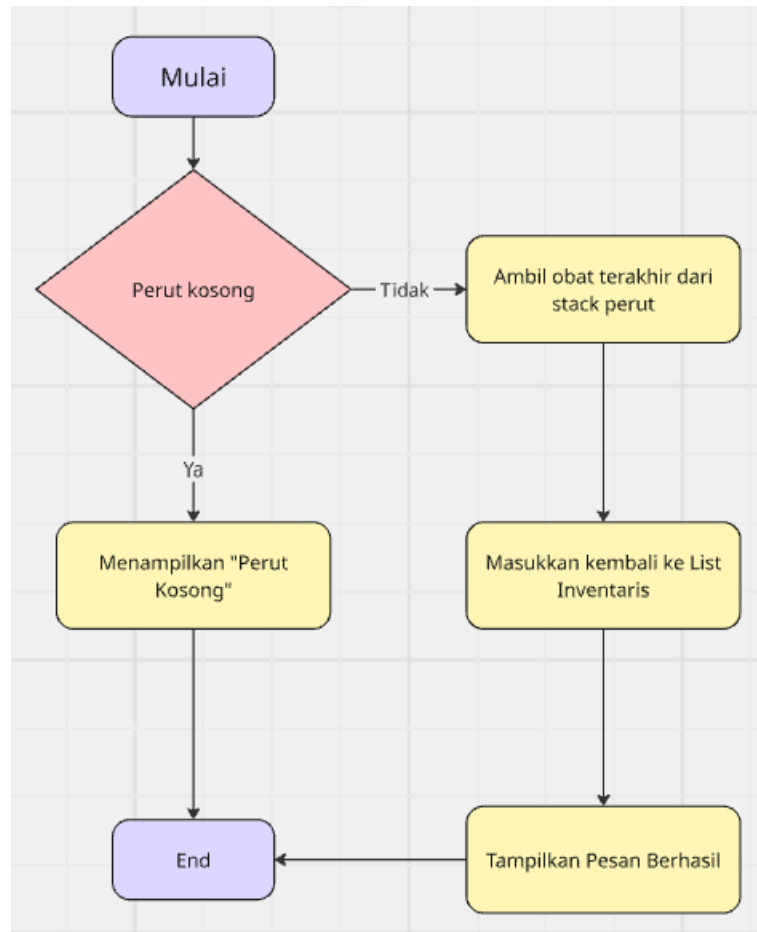
Gambar 6.16 Flowchart Antrian Saya!

16. F16 - Minum Obat



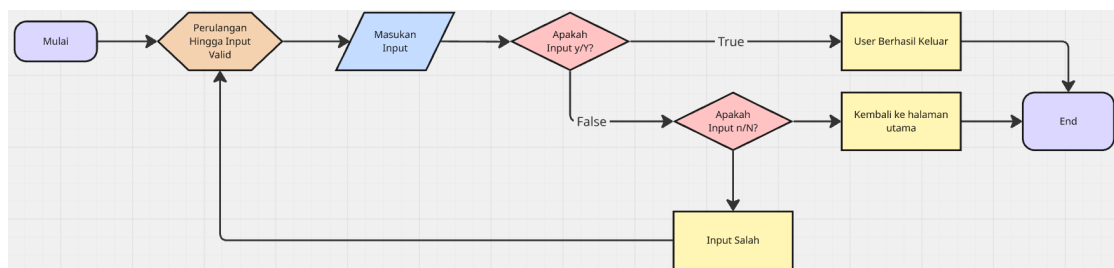
Gambar 6.17 Flowchart Minum Obat

17. F17 - Minum Penawar



Gambar 6.18 Flowchart Minum Penawar

18. F18 - Exit



Gambar 6.19 Flowchart Exit

SPESIFIKASI MODUL

1. Main Program

```
procedure tampilanBanner()
    output ""
    output "+-----+"
    output "|          SELAMAT DATANG          |"
    output "|          DI RUMAH SAKIT NIMONS    |"
    output "+-----+"
    output "                powered by Kelompok L"
    output "                K04 - STI"
    output ""

procedure tampilkanMenuUtama()
    if current_user = null then
        output ""
        output "Selamat datang! Silakan pilih opsi:"
        output "1. LOGIN"
        output "2. REGISTER PASIEN"
        output "4. LUPA PASSWORD"
        output "5. HELP"
        output "0. KELUAR"
    else
        output ""
        output "Halo, " + roleToStr(current_user.role) + " " +
current_user.username + "!"
        output "-----"
        output "Menu yang Tersedia:"
        output "1. LIHAT DENAH"
        output "2. LIHAT RUANGAN"
        output "5. HELP "

        if current_user.role = MANAGER then
            output "10. TAMBAH DOKTER"
            output "11. ASSIGN DOKTER"

            output "9. LOGOUT "
            output "0. EXIT"
        output "-----"
        output "Pilihan Anda: "

program utama
    tampilanBanner()

    initUserSet(generateDataPengguna)
    initMapDokterRuangan(generateDataMap)

    define jumlahRuangan, kapasitas as integer
    output "Masukkan jumlah ruangan: "
    input jumlahRuangan
    if input_gagal_atau_bukan_integer then
        output "Input tidak valid. Masukkan jumlah ruangan."
        → 1
```

```

clearInputBuffer()

output "Masukkan kapasitas ruangan: "
input kapasitas
if input_gagal_atau_bukan_integer then
    output "Input tidak valid. Masukkan kapasitas ruangan."
    → 1
clearInputBuffer()

initDenah(generateDataDenah, jumlahRuangan, kapasitas)

define pilihan as integer
define usernameInput as string[MAX_LEN]
define passwordInput as string[MAX_LEN]
define idRuanganInput as integer

repeat
    tampilkanMenuUtama()
    input pilihan
    if input_gagal_atau_bukan_integer then
        output "Input tidak valid. Masukkan nomor pilihan."
        clearInputBuffer()
        pilihan ← -1
        continue 1
    clearInputBuffer()

    if current_user = null then -- Belum Login
        switch pilihan
            case 1:
                output ">>> LOGIN"
                output "Username : "
                input usernameInput
                output "Password : "
                input passwordInput
                loginUser(generateDataPengguna, usernameInput,
passwordInput)
            case 2:
                output ">>> REGISTER"
                output "Username : "
                input usernameInput
                output "Password : "
                input passwordInput
                registerUser(generateDataPengguna, usernameInput,
passwordInput, PASIEN)
            case 4:
                lupaPassword(generateDataPengguna)
            case 5:
                output ">>> HELP"
                help()
            case 0:
                keluar()
            default:
                output "Pilihan tidak valid."

        else -- Sudah Login
            switch pilihan

```



```

        case 1:
            lihatDenah(generateDataDenah)
        case 2:
            output "Masukkan ID Ruangan yang ingin dilihat : "
            input idRuanganInput
            if input_berhasil_dan_integer then
                clearInputBuffer()
                lihatRuangan(generateDataDenah, idRuanganInput,
generateDataPengguna, generateDataMap)
            else
                output "Input ID Ruangan tidak valid."
                clearInputBuffer()
            endif
        case 5:
            output ">>> HELP"
            help()
        case 10:
            if current_user.role = MANAGER then
                managerTambahDokter(generateDataPengguna)
            else
                output "Anda tidak memiliki akses untuk fitur ini."
            endif
        case 11:
            if current_user.role = MANAGER then
                assignDokter(generateDataMap, generateDataDenah,
generateDataPengguna)
            else
                output "Anda tidak memiliki akses untuk fitur ini."
            endif
        case 9:
            logoutUser()
        case 0:
            keluar()
        default:
            output "Pilihan tidak valid."
    until pilihan = 0

    → 0

```

2. F01 - Login

```

function loginUser(set : UserSet, username : string, password :
string) → User
DEKLARASI
    user_found : User
    i : integer

DESKRIPSI
    input (username)
    input (password)
    user_found ← findUserByUsername(set, username)

    if user_found = NULL then
        output("Tidak ada Manager, Dokter, atau pun Pasien yang
bernama ", username)
        return NULL
    endif

```

```

endif

for i ← 0 HINGGA set.jumlahUser - 1 fo
    if set.users[i].username = username && set.users[i].password
= password then
        current_user ← set.users[i]
        output("Login berhasil. Halo, ", username)

        if current_user.role = MANAGER then
            output("Selamat pagi Manager ", current_user.username)
        else if current_user.role = DOKTER then
            output("Selamat pagi Dokter ", current_user.username)
        else if current_user.role = PASIEN then
            output("Selamat pagi Pasien ", current_user.username, "!
Ada keluhan apa?")
        endif

        return current_user
    endif
endfor

output("Login gagal. Username atau password salah.")
return NULL

```

3. F02 - Register

```

procedure registerUser (input username : string, input password :
string, input role : enum {MANAGER,DOKTER,PASIEN}, in/out set :
UserSet)

Deklarasi:
i : integer
usernameRegistered : boolean
newUser : User

For i ← 0 set.jumlahUser - 1 do
    if set.users[i].username = username then
        usernameRegistered ← true
        print "Username sudah terdaftar"
        return
    endif
endFor

if set.jumlahUser = MAX_USERS then
    print "User limit reached"
    return
endif

newUser.id ← set.jumlahUser + 1
newUser.username ← username
newUser.password ← password
newUser.role ← role

```

```
set.users[set.jumlahUser] ← newUser
set.jumlahUser ← set.jumlahUser + 1

Print "Registrasi berhasil untuk ", username
```

4. F03 - Logout

```
function logoutUser() → boolean

    output(">>> LOGOUT")

    if current_user != NULL then
        output("Logout berhasil. Sampai jumpa, ",
current_user.username)
        current_user ← NULL
        return true
    endif

    output("Belum login.")
    return false
```

5. F04 - Lupa Password

```
DEKLARASI
    username : string
    password : string
    role : string
    userCount : integer
    MAX_USERS : integer
    users : array of User

ALGORITMA
procedure registerUser(username : string, password : string, role
: string)
    if (userCount < MAX_USERS) then
        users[userCount].username ← username
        users[userCount].password ← password
        users[userCount].role ← role
        userCount ← userCount + 1
```

```
DEKLARASI
    username : string
    rleCode : string
    correctRle : string
    newPassword : string
    user : pointer to User

ALGORITMA
procedure lupaPassword()
    output(">>> LUPA_PASSWORD")
```

```

output("Username: ")
input(username)

user ← findUser(username)
if (user = NULL) then
    output("Username tidak terdaftar!")
    return

output("Kode Unik: ")
input(rleCode)

generateRLE(user.username, correctRle)

if (rleCode ≠ correctRle) then
    output("Kode unik salah!")
    return

output("Halo", user.role, user.username, ", silakan daftarkan
ulang password anda!")
output("Password Baru: ")
input(newPassword)

user.password ← newPassword
output("Password berhasil diubah!")

```

DEKLARASI

```

username : string
user : pointer to User
i : integer

```

ALGORITMA

```

procedure findUser(username : string) → pointer to User
    user ← NULL
    i ← 0

    while (i < userCount) do
        if (users[i].username = username) then
            user ← address of users[i]
            return user

        i ← i + 1

    return NULL

```

DEKLARASI

```

username : string
rleCode : string
len : integer
count : integer
i : integer
rleIndex : integer

```

ALGORITMA

```

procedure generateRLE(username : string, output rleCode : string)
    len ← panjang(username)

```

```

count ← 1
rleIndex ← 0

for i ← 1 to len do
    if (username[i] = username[i - 1]) and (i < len) then
        count ← count + 1
    else
        if count > 1 then
            append(count) to rleCode
        append(username[i - 1]) to rleCode
        count ← 1

append('\0') to rleCode

```

6. F05 - Menu & Help

```

procedure help

DEKLARASI
    current_user : Pointer to User
    username : string
    role : enum {MANAGER, DOKTER, PASIEN}

    if (current_user = NULL) then
        output("Kamu belum login sebagai role apapun. Silahkan
login terlebih dahulu.")
        output("")
        output("1. LOGIN: Masuk ke dalam akun yang sudah
terdaftar")
        output("2. REGISTER: Membuat akun baru")
    else
        case (current_user→role) of
            MANAGER:
                output("Halo Manager ", current_user→username, ".
Kenapa kamu memanggil command HELP? Kan kamu manager, tapi
yasudahlah kamu pasti sedang kebingungan. Berikut adalah hal-hal
yang dapat kamu lakukan:")
                output("1. LOGOUT: Keluar dari akun")
                output("2. TAMBAH_DOKTER: Menambahkan dokter
baru")
                output("3. LIHAT_USER: Melihat seluruh user")
                output("4. EXIT: Keluar dari sistem")

            DOKTER:
                output("Halo Dokter ", current_user→username, ".
Kamu memanggil command HELP. Kamu pasti sedang kebingungan.
Berikut adalah hal-hal yang dapat kamu lakukan:")
                output("1. LOGOUT: Keluar dari akun")
                output("2. DIAGNOSIS: Melakukan diagnosis
pasien")
                output("3. NGOBATIN: Memberikan obat")
                output("4. EXIT: Keluar dari sistem")

```

```

        PASIEN:
            output("Selamat datang, ", current_user->username,
". Kamu memanggil command HELP. Kamu pasti sedang kebingungan.
Berikut adalah hal-hal yang dapat kamu lakukan:")
            output("1. LOGOUT: Keluar dari akun")
            output("2. DAFTAR_CHECKUP: Daftar untuk
pemeriksaan")
            output("3. MINUM_OBAT: Minum obat")
            output("4. PULANGDOK: Meminta izin pulang")
            output("5. EXIT: Keluar dari sistem")

        output("")
        output("Footnote:")
        output("1. Untuk menggunakan aplikasi, silahkan masukkan nama
fungsi yang terdaftar")
        output("2. Jangan lupa untuk memasukkan input yang valid")

```

7. F06 - Denah Rumah Sakit

```

procedure initDenah (input/output denah: Denah*, input
jumlahRuangan : integer, input kapasitas : integer)

if denah = NULL then
    ->

if jumlahRuangan <0 OR kapasitas <0 then
    denah.jumlahRuangan <- 0
    ->

denah.jumlahRuangan <- jumlahRuangan
denah.kapasitas <- kapasiras

i traversal [0..jumlahRuangan - 1]
    denah.ruangan[i].idDokter <- ID_INVALID
    denah.ruangan[i].idRuangan <- ID_INVALID
    denah.ruangan[i].jumlahPasien <- 0
    j traversal {0..MAX_PASIEN-1} do
        denah.ruangan[i].idPasien[j] <- ID_INVALID
    initQueue(denah.ruangan[i].antrianPseien)

function getPanjangDenah(denah : Denah*) -> integer
if denah = NULL then
    -> 0
-> denah.jumlahRuangan

function getKapasitasDenah (denah: Denah*) -> integer
if denah = NULL then
    -> 0
-> denah.kapasitas

procedure lihatDenah(input/output denah : Denah*)
if denah = NULL or denah.jumlahRuangan = 0 then
    outout "tidak ada ruangan"

```

```

->

i traversal [0..denah.jumlahRuangan - 1]
  outout "+-----"

i traversal [0..denah.jumlahRuangan-1]
  output"| " + (i+1) " |"

i traversal [0..denah.jumlahRuangan - 1]
  outout "+-----"

procedure lihatRuangan(input/output denah: Denah*, input
idRuangan : integer, set : UserSet, map : mapDokterRuangan*)

idxRuangan ← idRuangan - 1
ruang ← denah.ruangan[idxRuangan]
output ">>> LIHAT_RUANGAN " + idRuangan
output ""
output "--- Detail Ruangan " + idRuangan + " ---"
output "Kapasitas : " + denah.kapasitas

idDokterDiRuangan ← getDokterByRuangan(map, idRuangan)
output "Dokter      : "
if idDokterDiRuangan ≠ ID_INVALID then
  dokter ← findUserById(set, idDokterDiRuangan)

  if dokter ≠ NULL then
    output "Dokter " + dokter.username
  else
    output "-"
ruangPasien ← denah.ruangan[idxRuangan]
outoput "Pasien di dalam ruangan : "
if ruangPasien.jumlahPasien > 0 then
  i traversal [0..ruangPasien.jumlahPasien - 1]
    pasien ← findUserById(set, ruangPasien.idPasien[i])
    if pasien ≠ NULL then
      outout " " + (i + 1) + ". " + pasien.username
else
  outout " Tidak ada pasien di dalam ruangan saat ini."
output "-----"

```

8. F07 - Lihat User

```

function lihatUser(set: UserSet)
DEKLARASI
  i : integer
DESKRIPSI
  output("Daftar Semua User:")
  for i ← 0 HINGGA set.jumlahUser - 1 do
    output("ID: ", set.users[i].id,
           ", Username: ", set.users[i].username,
           ", Role: ", set.users[i].role)
  endfor
endfunction

```

```

function lihatDokter(set: UserSet)
DEKLARASI
    i : integer
DESKRIPSI
    output("Daftar Dokter:")
    for i ← 0 HINGGA set.jumlahUser - 1 do
        if set.users[i].role = "DOKTER" then
            output("ID: ", set.users[i].id,
                ", Username: ", set.users[i].username)
        endif
    endfor
endfunction

function lihatPasien(set: UserSet)
DEKLARASI
    i : integer
DESKRIPSI
    output("Daftar Pasien:")
    for i ← 0 HINGGA set.jumlahUser - 1 do
        if set.users[i].role = "PASIEN" then
            output("ID: ", set.users[i].id,
                ", Username: ", set.users[i].username)
        endif
    endfor
endfunction

```

9. F08 - Cari User

```

function cariUser(set: UserSet, username: string) -> User
DEKLARASI
    i : integer
    user_found : User
    found : boolean
DESKRIPSI
    found ← false
    for i ← 0 HINGGA set.jumlahUser - 1 do
        if set.users[i].username = username then
            found ← true
            user_found ← set.users[i]
            break
        endif
    endfor

    if found then
        output("User ditemukan: ID: ", user_found.id,
            ", Username: ", user_found.username,
            ", Role: ", user_found.role)
        return user_found
    else
        output("User dengan username ", username, " tidak
ditemukan.")
        return NULL
    end

```



```

        endif
    endfunction

function cariDokter(set: UserSet, username: string) -> User
DEKLARASI
    i : integer
    user_found : User
    found : boolean
DESKRIPSI
    found ← false
    for i ← 0 HINGGA set.jumlahUser - 1 do
        if set.users[i].username = username && set.users[i].role
= "DOKTER" then
            found ← true
            user_found ← set.users[i]
            break
        endif
    endfor

    if found then
        output("Dokter ditemukan: ID: ", user_found.id,
            ", Username: ", user_found.username)
        return user_found
    else
        output("Dokter dengan username ", username, " tidak
ditemukan.")
        return NULL
    endif
endfunction

function cariPasien(set: UserSet, username: string) -> User
DEKLARASI
    i : integer
    user_found : User
    found : boolean
DESKRIPSI
    found ← false
    for i ← 0 HINGGA set.jumlahUser - 1 do
        if set.users[i].username = username && set.users[i].role
= "PASIEN" then
            found ← true
            user_found ← set.users[i]
            break
        endif
    endfor

    if found then
        output("Pasien ditemukan: ID: ", user_found.id,
            ", Username: ", user_found.username)
        return user_found
    else
        output("Pasien dengan username ", username, " tidak
ditemukan.")
        return NULL
    endif
endfunction

```

10. F09 - Lihat Antrian

```
procedure lihatSemuaAntrian (input denah : Denah*, input set :
UserSet*, input mapDokter : MapDokterRuangan*)
```

KAMUS LOKAL

```
    i : integer
    adaDetailRuanganDitampilkan : boolean
    ruangSaatIni : Ruangan*
    idRuanganSaatIni : integer
    idDokterDiRuangan : integer
    dokterInfo : User *
    nodeSaatIni : Node *
    pasienDiRuanganTampilCount : integer
    nomorUrutTampil : integer
    pasienInfo : User*
    adaPasienDiAntrianLuar : boolean
```

ALGORITMA

```
    if (denah = NULL OR set = NULL OR mapDokter = NULL) then
        ->
```

```
    output(">>> LIHAT SMEUA ANTRIAN)
```

```
    output ("Denah Ruangan:")
```

```
    lihatDenah(denah)
```

```
    if (denah.jumlahRuangan = 0) then
        output("Tidak ada ruangan yang tersedia")
        ->
```

```
    i traversal [0..denah.jumlahRuangan]
    ruang <- denah.ruangan[i]
    idRuangan <- ruang.idRuangan
    idDokter <- getDokterByRuangan(mapDokter, idRuangan)
```

```
    if ((idDokter != -1 OR !isEmpty(ruang.antrianPasien))
then
```

```
        adaDetailRuanganDitampilkan <- true
        output("=====", idRuanganSaatIni, "=====")
        output("Kapasitas  :", denah.kapasitas)
        { Tampilkan Dokter }
        output("Dokter      : ")
        if (idDokterDiRuangan != ID_INVALID) then
            dokterInfo <- findUserById(set, idDokterDiRuangan)
            if (dokterInfo != NULL) then
                output("Dr. ", dokterInfo.username)
            else
                output("-")
        else
            output("-")
```

```
        { Tampilkan Pasien di Dalam Ruangan }
```

```

        nodeSaatIni <- ruangSaatIni.antrianSemuaPasien.front
        output("")
        output("Pasien di dalam ruangan:")
        pasienDiRuanganTampilCount <- 0
        nomorUrutPasienDiRuangan <- 1
        while (nodeSaatIni != NULL AND pasienDiRuanganTampilCount
< denah.kapasitas) do
            pasienInfo <- findUserById(set,
nodeSaatIni.data.idPasien)
            if (pasienInfo != NULL) then
                output(" ", nomorUrutPasienDiRuangan, ". ",
pasienInfo.username)
            else
                output(" ", nomorUrutPasienDiRuangan, ". (Pasien
ID: ", nodeSaatIni.data.idPasien, " tidak ditemukan)")
                pasienDiRuanganTampilCount <-
pasienDiRuanganTampilCount + 1
                nomorUrutPasienDiRuangan <- nomorUrutPasienDiRuangan +
1
                nodeSaatIni <- nodeSaatIni.next

        if (pasienDiRuanganTampilCount = 0) then
            output(" Tidak ada pasien di dalam ruangan saat ini.")

        { Tampilkan Pasien di Antrian (sisanya dari queue) }
        output("Pasien di antrian:")
        nomorUrutPasienDiAntrianLuar <- 1
        adaPasienDiAntrianLuarTampil <- false
        while (nodeSaatIni != NULL) do
            pasienInfo <- findUserById(set,
nodeSaatIni.data.idPasien)
            if (pasienInfo != NULL) then
                output(" ", nomorUrutPasienDiAntrianLuar, ". ",
pasienInfo.username)
                adaPasienDiAntrianLuarTampil <- true
            else
                output(" ", nomorUrutPasienDiAntrianLuar, ".
(Pasien ID: ", nodeSaatIni.data.idPasien, " tidak ditemukan)")
                adaPasienDiAntrianLuarTampil <- true

                nomorUrutPasienDiAntrianLuar <-
nomorUrutPasienDiAntrianLuar + 1
                nodeSaatIni <- nodeSaatIni.next

        if (!adaPasienDiAntrianLuarTampil) then
            output(" Tidak ada pasien di antrian saat ini.")
            output("-----")

        if (!adaDetailRuanganDitampilkan AND denah.jumlahRuangan > 0)
then
            output("Semua ruangan kosong atau tidak memiliki dokter dan
antrian.")

```

11. F10 - Tambah Dokter

```
procedure initMapDokterRuangan(input/output map:
MapDokterRuangan*)
```

```
    if map = null then
        return
```

```
    map.jumlahDokter ← 0
    i traversal [0..MAX_DOKTER - 1]
        map.map[i].idDokter ← ID_INVALID
        map.map[i].idRuangan ← ID_INVALID
```

```
function assigDokterToRuangan(map: MapDokterRuangan*, idDokter:
integer, idRuangan: integer) -> boolean
```

```
    if map = null or idDokter = ID_INVALID or idRuangan =
ID_INVALID then
        -> false
```

```
    if map.jumlahDokter ≥ MAX_DOKTER then
        -> false
```

```
    if isDokterAssignedToRuangan(map, idDokter) then
        -> false
```

```
    if isRuanganAssignedTDokter(map, idRuangan)
        -> false
```

```
    map.map[map.jumlahDokter].idDokter ← idDokter
    map.map[map.jumlahDokter].idRuangan ← idRuangan
    map.jumlahDokter ← map.jumlahDokter + 1
```

```
    -> true
```

```
function unassignDokterFromRuangan( map: MapDokterRuanganPointer,
idDokter: integer) -> boolean
```

```
    if map = null or idDokter = ID_INVALID then
        -> false
```

```
    idxFound ← -1
    i travesak [0..map.jumlahDokter - 1]
        if map.map[i].idDokter = idDokter then
            idxFound ← i
            break
```

```
    if idxFound ≠ -1 then
        if idxFound < map.jumlahDokter - 1 then
            map.map[idxFound] ← map.map[map.jumlahDokter - 1]
            map.map[map.jumlahDokter - 1].idDokter ← ID_INVALID
            map.map[map.jumlahDokter - 1].idRuangan ← ID_INVALID
            map.jumlahDokter ← map.jumlahDokter - 1
            -> true
```

```
    else
        -> false
```

```

function getRuanganByDokter( map: MapDokterRuanganPointer,
idDokter: integer) -> integer
  if map = null or idDokter = ID_INVALID then
    -> ID_INVALID

  i traversal [0..map.jumlahDokter - 1]
  if map.map[i].idDokter = idDokter then
    -> map.map[i].idRuangan

  -> ID_INVALID

```

```

function getDokterByRuangan( map: MapDokterRuanganPointer,
idRuangan: integer) -> integer
  if map = null or idRuangan = ID_INVALID then
    -> ID_INVALID

  i traversal [0..map.jumlahDokter - 1 ]
  if map.map[i].idRuangan = idRuangan then
    -> map.map[i].idDokter

  -> ID_INVALID

```

```

function isDokterAssignedToRuangan( map: MapDokterRuanganPointer,
idDokter: integer) -> boolean
  -> getRuanganByDokter(map, idDokter) ≠ ID_INVALID

```

```

function isRuanganAssignedToDokter( map: MapDokterRuanganPointer,
idRuangan: integer) -> boolean
  -> getDokterByRuangan(map, idRuangan) ≠ ID_INVALID

```

```

procedure managerTambahDokter(input/output set: UserSetPointer)
  if set = null then
    →

  define username as string[MAX_LEN]
  define password as string[MAX_LEN]

  output ">>> TAMBAH_DOKTER"

  output "Username: "
  input username
  if input_gagal then
    output "Gagal membaca input."
    →

  existingUser ← findUserByUsername(set, username)
  if existingUser ≠ null then
    if existingUser.role = DOKTER then
      output "Username " + username + " sudah terdaftar sebagai
dokter."
    else
      output "Username " + username + " sudah terdaftar."
    →

  output "Password: "
  input password

```

```
if input_gagal then
    output "Gagal membaca input."
    →

dokterBaru ← registerUser(set, username, password, DOKTER)
```

```
procedure assignDokter(input/output map: MapDokterRuanganPointer,
input denah: DenahPointer, input set: UserSetPointer)
```

```
define username as string[MAX_LEN]
define idRuangan as integer
```

```
output ">>> ASSIGN_DOKTER"
output "Username: "
input username
if input_gagal then
    output "Gagal membaca input."
    →
```

```
output "ID Ruangan : "
input idRuangan
```

```
dokter ← findUserByUsername(set, username)
if dokter = null or dokter.role ≠ DOKTER then
    output "Dokter dengan username " + username + " tidak
ditemukan."
    →
```

```
if idRuangan < 1 or idRuangan > denah.jumlahRuangan then
    output "Ruangan dengan ID " + idRuangan + " tidak ditemukan."
    →
```

```
ruanganSekarang ← getRuanganByDokter(map, dokter.id)
dokterSekarangId ← getDokterByRuangan(map, idRuangan)
dokterDiRuanganItu ← findUserById(set, dokterSekarangId)
```

```
define namaDokterDiRuanganItu as string
if dokterDiRuanganItu ≠ null then
    namaDokterDiRuanganItu ← dokterDiRuanganItu.username
else
    namaDokterDiRuanganItu ← ""
```

```
if ruanganSekarang ≠ ID_INVALID and dokterSekarangId ≠
ID_INVALID then
    output ""
    output "Dokter " + dokter.username + " sudah ada di ruangan "
+ ruanganSekarang + "!"
    output "Ruangan " + idRuangan + " juga sudah ditempati dokter
" + namaDokterDiRuanganItu + "!"
    →
```

```
if ruanganSekarang ≠ ID_INVALID then
    output ""
    output "Dokter " + dokter.username + " sudah diassign ke
ruangan " + ruanganSekarang + "!"
    →
```

```

if dokterSekarangId ≠ ID_INVALID then
    output ""
    output "Dokter " + namaDokterDiRuanganItu + " sudah menempati
ruangan " + idRuangan + "!"
    output "Silakan cari ruangan lain untuk dokter " +
dokter.username + "."
    →

if assignDokterToRuangan(map, dokter.id, idRuangan) then
    output ""
    output "Dokter " + username + " berhasil diassign ke ruangan
" + idRuangan + "!"
else
    output ""
    output "Gagal mengassign dokter " + username + " ke ruangan "
+ idRuangan + "."

```

12. F11 - Diagnosis

```

procedure initPenyakit()
    jumlahPenyakit ← 0

    // Penyakit 1: Influenza
    if jumlahPenyakit < MAX_PENYAKIT then
        p ← daftarPenyakit[jumlahPenyakit]
        jumlahPenyakit ← jumlahPenyakit + 1

        p.id ← 1
        p.namaPenyakit ← "Influenza"
        p.suhu_min ← 36.0; p.suhu_max ← 38.5
        p.sistol_min ← 90; p.sistol_max ← 120
        p.diastol_min ← 60; p.diastol_max ← 80
        p.detak_min ← 60; p.detak_max ← 100
        p.saturasi_min ← 95.0; p.saturasi_max ← 100.0
        p.gula_min ← 70; p.gula_max ← 140
        p.berat_min ← 45.0; p.berat_max ← 90.0
        p.tinggi_min ← 150; p.tinggi_max ← 185
        p.kolesterol_min ← 150; p.kolesterol_max ← 240
        p.trombosit_min ← 150; p.trombosit_max ← 450

    // Penyakit 2: COVID 19
    if jumlahPenyakit < MAX_PENYAKIT then
        p ← daftarPenyakit[jumlahPenyakit]
        jumlahPenyakit ← jumlahPenyakit + 1

        p.id ← 2
        p.namaPenyakit ← "COVID 19"
        p.suhu_min ← 36.0; p.suhu_max ← 39.0
        p.sistol_min ← 90; p.sistol_max ← 130
        p.diastol_min ← 60; p.diastol_max ← 85
        p.detak_min ← 60; p.detak_max ← 120
        p.saturasi_min ← 90.0; p.saturasi_max ← 100.0

```

```

p.gula_min ← 70; p.gula_max ← 180
p.berat_min ← 45.0; p.berat_max ← 90.0
p.tinggi_min ← 150; p.tinggi_max ← 185
p.kolesterol_min ← 150; p.kolesterol_max ← 240
p.trombosit_min ← 150; p.trombosit_max ← 450

// Penyakit 3: Hipertensi
if jumlahPenyakit < MAX_PENYAKIT then
  p ← daftarPenyakit[jumlahPenyakit]
  jumlahPenyakit ← jumlahPenyakit + 1

  p.id ← 3
  p.namaPenyakit ← "Hipertensi"
  p.suhu_min ← 36.0; p.suhu_max ← 37.5
  p.sistol_min ← 140; p.sistol_max ← 180
  p.diastol_min ← 90; p.diastol_max ← 120
  p.detak_min ← 60; p.detak_max ← 100
  p.saturasi_min ← 90.0; p.saturasi_max ← 100.0
  p.gula_min ← 70; p.gula_max ← 140
  p.berat_min ← 45.0; p.berat_max ← 90.0
  p.tinggi_min ← 150; p.tinggi_max ← 185
  p.kolesterol_min ← 150; p.kolesterol_max ← 240
  p.trombosit_min ← 150; p.trombosit_max ← 450

// Penyakit 4: Diabetes Mellitus
if jumlahPenyakit < MAX_PENYAKIT then
  p ← daftarPenyakit[jumlahPenyakit]
  jumlahPenyakit ← jumlahPenyakit + 1

  p.id ← 4
  p.namaPenyakit ← "Diabetes Mellitus"
  p.suhu_min ← 36.0; p.suhu_max ← 37.5
  p.sistol_min ← 90; p.sistol_max ← 140
  p.diastol_min ← 60; p.diastol_max ← 90
  p.detak_min ← 60; p.detak_max ← 100
  p.saturasi_min ← 90.0; p.saturasi_max ← 100.0
  p.gula_min ← 126; p.gula_max ← 200
  p.berat_min ← 45.0; p.berat_max ← 90.0
  p.tinggi_min ← 150; p.tinggi_max ← 185
  p.kolesterol_min ← 150; p.kolesterol_max ← 240
  p.trombosit_min ← 150; p.trombosit_max ← 450

// Penyakit 5: Anemia
if jumlahPenyakit < MAX_PENYAKIT then
  p ← daftarPenyakit[jumlahPenyakit]
  jumlahPenyakit ← jumlahPenyakit + 1

  p.id ← 5
  p.namaPenyakit ← "Anemia"
  p.suhu_min ← 36.0; p.suhu_max ← 37.0
  p.sistol_min ← 90; p.sistol_max ← 120
  p.diastol_min ← 60; p.diastol_max ← 80
  p.detak_min ← 60; p.detak_max ← 100
  p.saturasi_min ← 95.0; p.saturasi_max ← 100.0
  p.gula_min ← 70; p.gula_max ← 140
  p.berat_min ← 45.0; p.berat_max ← 90.0

```



```
p.tinggi_min ← 150; p.tinggi_max ← 185
p.kolesterol_min ← 150; p.kolesterol_max ← 240
p.trombosit_min ← 150; p.trombosit_max ← 450
```

```
function getNamaPenyakit(id : integer)
  for i ← 0 until jumlahPenyakit - 1 do
    if daftarPenyakit[i].id = id then
      return daftarPenyakit[i].namaPenyakit
  return NULL
```

```
procedure diagnosisPasien(dokter : user, denah : denah, mapDokter
: mapdokterruangan, set : userset)
```

```
  if dokter == NULL and denah == NULL and mapDokter == NULL and
set == NULL then
    RETURN
```

```
  output ">>> DIAGNOSIS"
```

```
  idRuang ← getRuangByDokter(mapDokter, dokter.id)
```

```
  if idRuang == ID_INVALID or idRuang > denah.jumlahRuang
then
    output "Anda tidak memiliki ruangan yang valid!"
    RETURN
```

```
  ruangan ← denah.ruangan[idRuang - 1]
  antrian ← ruangan.antrianPasien
```

```
  if isEmpty(antrian) then
    output "Tidak ada pasien untuk diperiksa!"
    RETURN
```

```
  pasien ← antrian.front.data
  d ← pasien.dataCheckUp
  pasien.idPenyakit ← ID_INVALID
  namaPenyakit ← ""
```

```
  for i ← 0 until jumlahPenyakit - 1 do
    p ← daftarPenyakit[i]
```

```
    suhuCocok ← (d.suhuTubuh >= p.suhu_min and d.suhuTubuh <=
p.suhu_max)
    sistolCocok ← (d.tekananDarahSistol >= p.sistol_min and
d.tekananDarahSistol <= p.sistol_max)
    diastolCocok ← (d.tekananDarahDiastol >= p.diastol_min
and d.tekananDarahDiastol <= p.diastol_max)
    detakCocok ← (d.detakJantung >= p.detak_min and
d.detakJantung <= p.detak_max)
    saturasiCocok ← (d.saturasiOksigen >= p.saturasi_min and
d.saturasiCocok <= p.saturasi_max)
    gulaCocok ← (d.kadarGulaDarah >= p.gula_min and
d.kadarGulaDarah <= p.gula_max)
    beratCocok ← (d.beratBadan >= p.berat_min and
d.beratBadan <= p.berat_max)
    tinggiCocok ← (d.tinggiBadan >= p.tinggi_min and
```

```

d.tinggiBadan <= p.tinggi_max)
    kolesterolCocok ← (d.kadarKolesterolTotal >=
p.kolesterol_min and d.kadarKolesterolTotal <= p.kolesterol_max)
    kolesterolLDLCocok ← (d.kadarKolesterolLDL >=
p.kolesterol_min and d.kadarKolesterolLDL <= p.kolesterol_max)
    trombositCocok ← (d.trombosit >= p.trombosit_min and
d.trombosit <= p.trombosit_max])

    if suhuCocok and sistolCocok and diastolCocok and
detakCocok and saturasiCocok and gulaCocok and beratCocok and
tinggiCocok and kolesterolCocok and kolesterolLDLCocok and
trombositCocok then

        pasien.idPenyakit ← p.id
        namaPenyakit ← p.namaPenyakit

    info ← findUserById(set, pasien.idPasien)

    if info ≠ NULL then
        if pasien.idPenyakit ≠ ID_INVALID then
            output "<username> terdiagnosa penyakit
<namaPenyakit>!"
        else
            output "<username> tidak terdiagnosis penyakit
apapun!"

```

13. F12 - Ngobatin

```

procedure initObat()
    jumlahObat ← 0

    // Obat 1
    if jumlahObat < MAX_OBAT then
        o ← daftarObat[jumlahObat]
        o.idObat ← 1
        o.namaObat ← "Oseltamivir"
        jumlahObat ← jumlahObat + 1

    // Obat 2
    if jumlahObat < MAX_OBAT then
        o ← daftarObat[jumlahObat]
        o.idObat ← 2
        o.namaObat ← "Vitamin C"
        jumlahObat ← jumlahObat + 1

    // Obat 3
    if jumlahObat < MAX_OBAT then
        o ← daftarObat[jumlahObat]
        o.idObat ← 3
        o.namaObat ← "Remdesivir"
        jumlahObat ← jumlahObat + 1

    // Obat 4

```

```

if jumlahObat < MAX_OBAT then
    o ← daftarObat[jumlahObat]
    o.idObat ← 4
    o.namaObat ← "Lisinopril"
    jumlahObat ← jumlahObat + 1
ENDIF

// Obat 5
if jumlahObat < MAX_OBAT then
    o ← daftarObat[jumlahObat]
    o.idObat ← 5
    o.namaObat ← "Metformin"
    jumlahObat ← jumlahObat + 1

jumlahMapObat ← 0

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 1
    map.idObat ← 1
    map.urutan ← 1

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 1
    map.idObat ← 2
    map.urutan ← 2

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 2
    map.idObat ← 3
    map.urutan ← 1

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 3
    map.idObat ← 4
    map.urutan ← 1

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 4
    map.idObat ← 5
    map.urutan ← 1

if jumlahMapObat < MAX_OBAT then
    map ← daftarMapObat[jumlahMapObat]
    jumlahMapObat ← jumlahMapObat + 1
    map.idPenyakit ← 5
    map.idObat ← 3

```

```

        map.urutan ← 1

    if jumlahMapObat < MAX_OBAT then
        map ← daftarMapObat[jumlahMapObat]
        jumlahMapObat ← jumlahMapObat + 1
        map.idPenyakit ← 5
        map.idObat ← 4
        map.urutan ← 3

    if jumlahMapObat < MAX_OBAT then
        map ← daftarMapObat[jumlahMapObat]
        jumlahMapObat ← jumlahMapObat + 1
        map.idPenyakit ← 5
        map.idObat ← 5
        map.urutan ← 2

```

```

function getNamaObat(id : integer)
    for i ← 0 until jumlahObat - 1 do
        if daftarObat[i].idObat = id then
            return daftarObat[i].namaObat
    return NULL

```

```

procedure berobatPasien(dokter, denah, mapDokter, set)
    if dokter == NULL and denah == NULL and mapDokter == NULL
and set == NULL then
        RETURN

    output ">>> BEROBAT"
    output "Dokter sedang mengobati pasien!"

    idRuangan ← getRuanganByDokter(mapDokter, dokter.id)
    if idRuangan == ID_INVALID or idRuangan > denah.jumlahRuangan
then RETURN

    ruangan ← denah.ruangan[idRuangan - 1]
    antrian ← ruangan.antrianPasien

    if isEmpty(antrian) then
        output "Tidak ada pasien untuk diperiksa!"
        RETURN

    if not dequeue(antrian, pasien) then
        output "Gagal mendapatkan pasien dari antrian!"
        RETURN

    info ← findUserById(set, pasien.idPasien)
    if infoPasien == NULL then
        output "Pasien tidak ditemukan!"
        RETURN

    if pasien.idPenyakit = -1 then
        output "Pasien tidak memiliki penyakit!"
        output "Pasien belum di diagnosis!"
        RETURN

```

```

else
    namaPenyakit ← getNamePenyakit(pasien.idPenyakit)
    if infoPasien != NULL and namaPenyakit != NULL then
        output "Pasien <username> memiliki penyakit
<namaPenyakit>."
        output "Obat yang harus diberikan:"

        adaObat ← FALSE
        for urutan ← 1 until MAX_OBAT do
            for i ← 0 until jumlahMapObat - 1 do
                if daftarMapObat[i].idPenyakit ==
pasien.idPenyakit and daftarMapObat[i].urutan == urutanObat THEN
                    adaObat ← TRUE
                    output urutan + ". " +
getNameObat(map.idObat)
                    BREAK

            if not adaResepObat then
                output "Tidak ada obat yang diresepkan untuk penyakit
ini."
        if infoPasien != NULL
            if (pasien.idPenyakit != ID_INVALID)
                output "Pasien <username> telah selesai berobat."
            else
                output "Pasien <username> tidak memiliki penyakit
yang terdiagnosis"

```

14. F13 - Aku boleh pulang ga, dok?

```

procedure F13_PulangDok(pasien_login : User*, denah : Denah*)

DEKLARASI
    entriPasien : pointer to EntriAntrianCheckup
    resepPenyakit : array of integer
    jumlahResep : integer
    i : integer
    valid : boolean

ALGORITMA
    output ">>> AKU BOLEH PULANG GA, DOK?"

    entriPasien ← getEntriPasien(denah, pasien_login.id)

    if entriPasien = NULL then
        output "Kamu tidak sedang mengantri check-up di rumah
sakit!"
        return

    if entriPasien.idPenyakit = ID_INVALID then
        output "Kamu belum didiagnosis! Konsultasikan dulu dengan
dokter."
        return

```

```

jumlahResep ← getJumlahResep(entriPasien.idPenyakit)

for i ← jumlahResep - 1 downto 0 do
    obatResep ← getObatKe(entriPasien.idPenyakit, i)
    obatPerut ← peekStack(pasien_login.perut_pasien)

    if obatResep ≠ obatPerut then
        output "Kamu salah urutan minum obat. Coba muntahkan
dulu dengan PENAWAR."
        return

    popStack(pasien_login.perut_pasien)

output "Selamat! Kamu sudah boleh pulang. Jaga kesehatan, ya!"

dequeuePasien(denah, pasien_login.id)

```

15. F14 - Daftar Check-Up

```

function isPasienTerdaftar (denah : Denah, idPasien: integer) ->
boolean
    if (denah = NULL and idPasien <= 0) then
        -> false
    i traversal [0..denah.jumlahRuangan]
        if (getPositionPasien(denah.ruangan[i].antrianPasien,
idPasien) != -1) then
            -> true

    -> false

```

```

procedure daftarCheckUp (input/output set: UserSet, input/output
denah: Denah, input/output mapDokter: MapDokterRUangan,
input/output currentUser: User)

```

KAMUS LOKAL

```

dataInput : DataCheckUp
suhuTubuh, saturasi, beratBadan, tinggiBadan : float
sistol, diastol, detakJantung, kadarGula, kolesterolTotal,
kolesterolLDL, trombosit : integer
dokterTersediaCount : integer
listDokterTersediaID : array of integer [0..MAX_DOKTER-1]
listRuanganTersediaID : array of integer [0..MAX_DOKTER-1]
i : integer {indeks iterasi}
dokterLoop : User
idRuanganDokter, idRuanganYangDipilih : integer
ruanganTarget, ruanganUntukEnqueue : Ruangan*
jumlahAntrianPasien : integer
pilihanDokterNomor : integer
buffPilihan : string
infoDokterDipilih : User*
entriPasienBaru : EntriAntrianCheckup

```

ALGORITMA

```
    if (current_user = NULL) then
        output("Anda harus login terlebih dahulu")
        ->
    if (current_user.role != PASIEN) then
        output("Hanya pasien yang dapat melakukan pendaftaran
check-up)
        ->
    output(">>> DAFTAR CHECKUP)

    {kasus 2}
    if (isPasienTerdaftar(denah, current_user.id)) then
        output("Anda sudah terdaftar dalam antrian checkup")
        output("Silakan selesaikan check-up yang sudah terdaftar
terlebih dahulu")
        ->

    {kasus 1 3}
    output("Silakan masukkan data cek-up Anda")
    dataInput <- createDataCheckUp(current_user.id)
    suhuTubuh <- getPositiveFloat("Suhu Tubuh (Celcius)")
    if (suhuTubuh = -1.0) then ->

    getPositiveTekananDarah("Tekanan Darah (Sistol/Diastol, contoh
120 80)", sistol, diastol)
    if (sistol = -1 atau diastol = -1) then ->

    detakJantung <- getPositiveInt("Detak Jantung (bpm)")
    if (detakJantung = -1) then ->

    saturasi <- getPositiveFloat("Saturasi Oksigen (%)")
    if (saturasi = -1.0) then ->

    kadarGula <- getPositiveInt("Kadar Gula Darah (mg/dL)")
    if (kadarGula = -1) then ->

    beratBadan <- getPositiveFloat("Berat Badan (kg)")
    if (beratBadan = -1.0) then ->

    tinggiBadanInt <- getPositiveInt("Tinggi Badan (cm)") }
    if (tinggiBadanInt = -1) then ->

    kolesterolTotal <- getPositiveInt("Kadar Kolesterol Total
(mg/dL)")
    if (kolesterolTotal = -1) then ->

    kolesterolLDL <- getPositiveInt("Kadar Kolesterol LDL
(mg/dL)")
    if (kolesterolLDL = -1) then ->

    trombosit <- getPositiveInt("Trombosit (ribu/ $\mu$ L)")
    if (trombosit = -1) then ->

    setDataCheckUp(&dataInput, suhuTubuh, sistol, diastol,
```

```

detakJantung, saturasi, kadarGula, beratBadan, tinggiBadan,
kolesterolTotal, kolesterolLDL, trombosit)

{ Menampilkan Dokter yang Tersedia }
output("") {Newline}
output("Berikut adalah daftar dokter yang tersedia:")
dokterTersediaCount <- 0
i traversal [0..set.jumlahUser]
  if (set.users[i].role = DOKTER) then
    dokter <- set.users[i]
    idRuanganDokter <- getRuanganByDokter(mapDokter,
dokter.id)
    if (idRuanganDokter != ID_INVALID AND idRuanganDokter >
0 AND idRuanganDokter <= denah.jumlahRuangan) then
      dokterTersediaCount <- dokterTersediaCount + 1
      if (dokterTersediaCount <= MAX_DOKTER) then
        listDokterTersediaID[dokterTersediaCount - 1] <-
dokter.id
        listRuanganTersediaID[dokterTersediaCount - 1] <-
idRuanganDokter
        ruanganDokter <- denah.ruangan[idRuangan-1]
        jumlahAntrianDiRuangan <-
getSize(ruanganTarget.antrianSemuaPasien)
        output(dokterTersediaCount, ". Dr. ",
dokterLoop.username, " - Ruangan ", idRuanganDokter, " (Total
Antrian: ", jumlahAntrianDiRuangan, " orang)")
      else
        break

  if (dokterTersediaCount = 0) then
    output("Maaf, tidak ada dokter yang tersedia saat ini.")
    ->

{ Pilih Dokter }
do
  output("") {Newline}
  output("Pilih dokter (1-", dokterTersediaCount, "): ")
  input(buffPilihan)
  if (inputGagal()) then
    output("Gagal membaca input.")
    ->
  pilihanDokterNomor <- stringKeInteger(buffPilihan)
  if (konversiBerhasil() dan pilihanDokterNomor >= 1 dan
pilihanDokterNomor <= dokterTersediaCount) then
    break
  else
    output("Input tidak valid. Silakan pilih dokter yang
tersedia.")
    while (true)

      idRuanganYangDipilih <-
listRuanganTersediaID[pilihanDokterNomor - 1]
      infoDokterDipilih <- findUserById(set,
listDokterTersediaID[pilihanDokterNomor - 1])
      ruanganDipilih <- &(denah.ruangan[idRuanganYangDipilih - 1])

```



```

    entriPasienBaru.idPasien <- currentUser.id
    entriPasienBaru.nomorAntri <-
    getSize(ruanganUntukEnqueue.antrianPasien) + 1
    entriPasienBaru.dataCheckUp <- dataInput
    entriPasienBaru.idPenyakit <- ID_INVALID

    if enqueue(ruanganDipilih.antrianPasien, entriPasien) then
        output("") {Newline}
        output("Pendaftaran check-up berhasil!")
        namaDokterStr <- "-"
        if (infoDokterDipilih != NULL) then
            namaDokterStr <- "Dr. " + infoDokterDipilih.username

        namaRuanganStr <- integerKeString(idRuanganYangDipilih)
    {Atau format lain}
        output("Anda terdaftar pada antrian ", namaDokterStr, " di
    ruangan ", namaRuanganStr, ".")
        output("Posisi antrian Anda: ", entriPasienBaru.nomorAntri)
    else
        output("Gagal mendaftar ke dalam antrian.")

```

16. F15 - Antrian Saya!

```

procedure lihatStatusAntrian(set : pointer to UserSet,denah :
pointer to Denah, mapDokter : pointer to MapDokterRuangan,
currentUser : pointer to User)

KAMUS LOKAL
    found: boolean
    i, posisi, total, kapasitas, idDokter: integer
    antrian: pointer to Queue
    dokter: pointer to User

ALGORITMA
    if (currentUser = NULL) or (set = NULL) or (denah = NULL)
or (mapDokter = NULL) then
        output("Ndak iso :")
        return

    if (currentUser->role ≠ PASIEN) then
        output("Selain pasien tidak bisa melihat status antrian")
        return

    output(">>> ANTRIAN\n")
    found ← false

    i traversal [0 ... denah->jumlahRuangan - 1]
        antrian ← address of denah->ruangan[i].antrianPasien

        if (isEmpty(antrian)) then
            continue

```

```

        posisi ← getPositionPasien(antrian, currentUser->id)

        if (posisi ≠ -1) then
            total ← getSize(antrian)
            kapasitas ← denah->kapasitas
            idDokter ← getDokterByRuangan(mapDokter,
denah->ruangan[i].idRuangan)
            dokter ← findUserById(set, idDokter)

            output("Status antrian Anda:")
            output("Dokter: " +
                if (dokter ≠ NULL and dokter->role = DOKTER)
then "Dr. " else "" +
                if (dokter ≠ NULL) then dokter->username else
"???)")

            output("Ruangan: " + denah->ruangan[i].idRuangan)

            if (posisi ≤ kapasitas) then
                output("Anda sedang berada di dalam ruangan
dokter!")
            else
                output("Posisi antrian: " + (posisi - kapasitas)
+ " dari " + (total - kapasitas))

            found ← true
            break

        if (not found) then
            output("Anda belum terdaftar dalam antrian check-up!")
            output("Silakan daftar terlebih dahulu dengan command
DAFTAR_CHECKUP.")

```

17. F16 - Minum Obat

```

ALGORITMA
    output(">>> MINUM_OBAT")
    if (pasien_login.inventory_obat.jumlah = 0) then
        output("Inventaris obat Anda kosong!")
        -> // Keluar dari prosedur
    endif

    output("===== DAFTAR OBAT DI INVENTORY =====")
    for i <- 0 to pasien_login.inventory_obat.jumlah - 1 do
        nama_obat <-
getNamaObat(pasien_login.inventory_obat.data[i])
        output(i+1, ". ", nama_obat)

    endfor

    output("Pilih obat untuk diminum: ")
    input(pilihan)

```

```

        if (pilihan < 1 OR pilihan >
pasien_login.inventory_obat.jumlah) then
            output("Pilihan nomor tidak tersedia!")
            ->
        endif

        obat_dipilih_id <-
pasien_login.inventory_obat.data[pilihan-1]

        for i <- pilihan-1 to pasien_login.inventory_obat.jumlah - 2
do
            pasien_login.inventory_obat.data[i] <-
pasien_login.inventory_obat.data[i+1]
            endfor

        pasien_login.inventory_obat.jumlah <-
pasien_login.inventory_obat.jumlah - 1

        pasien_login.perut_pasien.top <-
pasien_login.perut_pasien.top + 1
        pasien_login.perut_pasien.data[pasien_login.perut_pasien.top]
<- obat_dipilih_id

        nama_obat <- getNamaObat(obat_dipilih_id)
        output("GLEKGLEKGLEK... ", nama_obat, " berhasildiminum!!!")

```

18. F17 - Minum Penawar

```

ALGORITMA
    output(">>> PENAWAR")
    if (pasien_login.perut_pasien.top = -1) then
        output("Perut kosong!! Belum ada obat yang dimakan.")
        ->
    endif

    obat_dikeluarkan_id <-
pasien_login.perut_pasien.data[pasien_login.perut_pasien.top]

    pasien_login.perut_pasien.top <-
pasien_login.perut_pasien.top - 1

    if (pasien_login.inventory_obat.jumlah < MAX_OBAT) then
pasien_login.inventory_obat.data[pasien_login.inventory_obat.jumla
h] <- obat_dikeluarkan_id

        pasien_login.inventory_obat.jumlah <-
pasien_login.inventory_obat.jumlah + 1

```

```
        nama_obat <- getNamaObat(obat_dikeluarkan_id)
        output("Uwekkk!!! ", nama_obat, " keluar dan kembali ke
inventory")
    else
        output("Inventaris penuh! Obat tidak bisa dikembalikan.")
    endif
```

19. F18 - Exit

```
procedure keluar (input: char)

    repeat
        output("Apakah Anda sudah selesai menggunakan layanan ini?
(y/n) ")
        input(input)

        if input = 'y' or input = 'Y' then
            output("Terima kasih telah menggunakan sistem rumah sakit
Nimons. Sampai jumpa!")
            stop

        else if input = 'n' or input = 'N' then
            output("Terima kasih telah menggunakan sistem rumah sakit
Nimons. Sampai jumpa!")
            stop

        else
            output("Input tidak valid. Harap masukkan 'y' atau 'n'.")
        endif

    until FALSE
```

HASIL PENGUJIAN PROGRAM

1. Inisialisasi Main Program

```
+-----+
|               SELAMAT DATANG               |
|               DI RUMAH SAKIT NIMONS         |
+-----+
              powered by Kelompok L
              K04 - STI

Masukkan jumlah ruangan: 5
Masukkan kapasitas ruangan: 3

Selamat datang! Silakan pilih opsi:
1. LOGIN
2. REGISTER PASIEN
4. LUPA PASSWORD
5. HELP
0. KELUAR
-----
```

Gambar 7.1 Inisialisasi Main Program

2. F01 - Login

#Kasus 1

Login sebagai Manager

Didefinisikan ketika inisialisasi list User langsung tercatat satu akun dengan role MANAGER username “admin” dan password “admin123”

```
-----
Pilihan Anda: 1
>>> LOGIN
Username : admin
Password : admin123
Login berhasil. Halo, admin!
Selamat pagi Manager admin!
```

Gambar 7.2 Login sebagai Manager

#Kasus 2

Login sebagai Dokter

Asumsi telah melakukan command TAMBAH DOKTER sebelum melakukan login

```
Pilihan Anda: 1
>>> LOGIN
Username : nico
Password : nico123
Login berhasil. Halo, nico!
Selamat pagi Dokter nico!

Halo, DOKTER nico!
-----
```

Gambar 7.3 Login sebagai Dokter

#Kasus 3

Login sebagai Pasien

Asumsi telah melakukan register

```
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami123
Login berhasil. Halo, ami!
Selamat pagi Pasien ami! Ada keluhan apa?

Halo, PASIEN ami!
-----
```

Gambar 7.4 Login sebagai Pasien

#Kasus 4

Tidak ada username terdaftar

```
Pilihan Anda: 1
>>> LOGIN
Username : ngarang
Password : ngarang123
Tidak ada Manager, Dokter, atau pun Pasien yang bernama ngarang!
```

Gambar 7.5 Tidak Ada Username Terdaftar

#Kasus 5

Kredensial salah

```
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami456
Login gagal. Username atau password salah.
```

Gambar 7.6 Kredensial Salah

3. F02 - Register

#Kasus 1

Register berhasil

```
>>> REGISTER
Username : ami
Password : ami123
Registrasi berhasil untuk ami.

Selamat datang! Silakan pilih opsi:
1. LOGIN
2. REGISTER PASIEN
4. LUPA PASSWORD
5. HELP
0. KELUAR
-----
Pilihan Anda: 1
>>> LOGIN
Username : ami
Password : ami123
Login berhasil. Halo, ami!
Selamat pagi Pasien ami! Ada keluhan apa?
```

Gambar 7.7 Register Berhasil

#Kasus 2

Register gagal

```
Pilihan Anda: 2
>>> REGISTER
Username : ami
Password : amilagi
Username ami sudah terdaftar.
```

Gambar 7.8 Register Gagal

4. F03 - Logout

#Kasus 1

Logout berhasil

```
>>> LOGIN
Username : admin
Password : admin123
Login berhasil. Halo, admin!
Selamat pagi Manager admin!

Halo, MANAGER admin!
-----
Menu yang Tersedia:
1. LIHAT DENAH
2. LIHAT RUANGAN
5. HELP
10. TAMBAH DOKTER
11. ASSIGN DOKTER
9. LOGOUT
0. EXIT
-----
Pilihan Anda: 9
>>> LOGOUT
Logout berhasil. Sampai jumpa, admin!
```

Gambar 7.9 Logout

5. F04 - Lupa Password

#Kasus 1

Ubah password berhasil

```
Testcase 1: Jeffreey
>>> LUPA_PASSWORD
Username: Jeffreey
Kode Unik: Je2fr2ey
Halo Dokter Jeffreey, silakan daftarkan ulang password anda!
Password Baru: JR1234
Password berhasil diubah!
>>> LOGIN
Username: Jeffreey
Password: JR1234
Selamat pagi Dokter Jeffreey!
```

Gambar 7.10 Password Berhasil Diubah

#Kasus 2
User tidak terdaftar

```
Testcase 2: NONEXISTENT_USER
>>> LUPA_PASSWORD
Username: NONEXISTENT_USER
Username tidak terdaftar!
```

Gambar 7.11 User Tidak Terdaftar

#Kasus 3
Kode unik tidak sesuai

```
Testcase 3: nimonsslatte
>>> LUPA_PASSWORD
Username: nimonsslatte
Kode Unik: nimonsslatte
Kode unik salah!
```

Gambar 7.12 Kode Unik Salah

6. F05 - Menu Help

#Kasus 1
Help belum login

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. LOGIN: Masuk ke dalam akun yang sudah terdaftar
2. REGISTER: Membuat akun baru

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.13 Help Belum Login

#Kasus 2

Help Manager

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Halo Manager admin. Kenapa kamu memanggil command HELP? Kan kamu manager, tapi yasudahla
h kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. TAMBAH_DOKTER: Menambahkan dokter baru
3. LIHAT_USER: Melihat seluruh user
4. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.14 Help Manager

#Kasus 3

Help Dokter

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Halo Dokter nico. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut ad
alah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. DIAGNOSIS: Melakukan diagnosis pasien
3. NGOBATIN: Memberikan obat
4. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.15 Help Dokter

#Kasus 4

Help Pasien

```
Pilihan Anda: 5
>>> HELP
===== HELP =====
Selamat datang, ami. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut
adalah hal-hal yang dapat kamu lakukan:
1. LOGOUT: Keluar dari akun
2. DAFTAR_CHECKUP: Daftar untuk pemeriksaan
3. MINUM_OBAT: Minum obat
4. PULANGDOK: Meminta izin pulang
5. EXIT: Keluar dari sistem

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.16 Help Pasien

7. F06 - Lihat Denah

Inisialisasi jumlah ruangan dan kapasitas tiap ruangan

```
+-----+
|               |
|   SELAMAT DATANG   |
|   DI RUMAH SAKIT NIMONS   |
|               |
+-----+
```

powered by Kelompok L
K04 - STI

Masukkan jumlah ruangan: 5
Masukkan kapasitas ruangan: 3

Pilihan Anda: 1

>>> LIHAT_DENAH

```
+-----+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |  5  |
+-----+-----+-----+-----+-----+
```

Gambar 7.17 Lihat Denah

```
-----
Pilihan Anda: 2
Masukkan ID Ruangan yang ingin dilihat : 1
>>> LIHAT_RUANGAN 1

--- Detail Ruangan 1 ---
Kapasitas : 3
Dokter    : Dokter gracie

Pasien di dalam ruangan :
    Tidak ada pasien di dalam ruangan saat ini.
-----
```

Gambar 7.18 Lihat Ruangan

8. F07 - Lihat User

```
=== MENU MANAGER ===
1. LIHAT_USER
2. LIHAT_PASIEN
3. LIHAT_DOKTER
4. CARI_USER
5. CARI_PASIEN
6. CARI_DOKTER
0. Keluar
>>> Pilihan: 1
Sort berdasarkan?
1. ID
2. Nama
>>> 1
Urutan?
1. ASC
2. DESC
>>> 1
  ID  Username  Role  Penyakit
  1   alice    Dokter -
  2   bob      Pasien Influenza
  3   carol    Dokter -
  4   dave     Pasien COVID 19

=== MENU MANAGER ===
1. LIHAT_USER
2. LIHAT_PASIEN
3. LIHAT_DOKTER
4. CARI_USER
5. CARI_PASIEN
6. CARI_DOKTER
0. Keluar
>>> Pilihan: 2
Sort berdasarkan?
1. ID
2. Nama
>>> 1
Urutan?
1. ASC
2. DESC
>>> 1
  ID  Username  Role  Penyakit
  2   bob      Pasien Influenza
  4   dave     Pasien COVID 19
```

Gambar 7.19 Lihat List User

9. F08 - Cari User

```
=== MENU MANAGER ===
1. LIHAT_USER
2. LIHAT_PASIEN
3. LIHAT_DOKTER
4. CARI_USER
5. CARI_PASIEN
6. CARI_DOKTER
0. Keluar
>>> Pilihan: 4
Cari berdasarkan:
1. ID
2. Nama
>>> 1
Masukkan ID: 1
  ID   Username   Role   Penyakit
  1    alice     Dokter -

=== MENU MANAGER ===
1. LIHAT_USER
2. LIHAT_PASIEN
3. LIHAT_DOKTER
4. CARI_USER
5. CARI_PASIEN
6. CARI_DOKTER
0. Keluar
>>> Pilihan: 5
Cari pasien berdasarkan:
1. ID
2. Nama
3. Penyakit
>>> 3
Masukkan nama penyakit: COVID 19
  ID   Username   Role   Penyakit
  4    dave      Pasien COVID 19
```

Gambar 7.20 Cari User dan Pasien

10. F09 - Lihat Antrian

```
Pilihan Anda: 5
>>> LIHAT_SEMUA_ANTRIAN

Denah Ruangan:
>>> LIHAT_DENAH
+---+---+
| 1 | 2 |
+---+---+

=====1=====
Kapasitas : 1
Dokter    : Dr. jack

Pasien di dalam ruangan:
  1. mi

Pasien di antrian:
  1. dif
-----

=====2=====
Kapasitas : 1
Dokter    : Dr. andrew

Pasien di dalam ruangan:
  1. van

Pasien di antrian:
  Tidak ada pasien di antrian saat ini.
-----
```

Gambar 7.21 Lihat Antrian

11. F10 - Tambah Dokter

#Kasus 1

Berhasil tambah dokter oleh manager

```
Halo, MANAGER admin!
-----
Menu yang Tersedia:
1. LIHAT DENAH
2. LIHAT RUANGAN
5. HELP
10. TAMBAH DOKTER
11. ASSIGN DOKTER
9. LOGOUT
0. EXIT
-----

Pilihan Anda: 10
>>> TAMBAH_DOKTER
Username: gracie
Password: gracie123
Registrasi berhasil untuk gracie.
Username gracie sudah terdaftar.
```

Gambar 7.22 Berhasil Tambah Dokter

#Kasus 2

Tidak berhasil tambah dokter oleh manager

```
Pilihan Anda: 10
>>> TAMBAH_DOKTER
Username: gracie
Username gracie sudah terdaftar sebagai dokter.
```

Gambar 7.23 Tidak Berhasil Tambah Dokter

#Kasus 1

Berhasil assign dokter ke ruangan

```
Pilihan Anda: 11
>>> ASSIGN_DOKTER
Username: gracie
ID Ruangan : 1

Dokter gracie berhasil diassign ke ruangan 1!
```

Gambar 7.24 Berhasil Assign Dokter ke Ruangan

#Kasus 2

Dokter sudah diassign ke ruangan lain sebelumnya

```
-----
Pilihan Anda: 11
>>> ASSIGN_DOKTER
Username: gracie
ID Ruangan : 2

Dokter gracie sudah diassign ke ruangan 1!
```

Gambar 7.25 Dokter Sudah Diassign ke Ruangan Lain

#Kasus 3

Ruangan sudah ditempati

```
Pilihan Anda: 11
>>> ASSIGN_DOKTER
Username: dina
ID Ruangan : 1

Dokter gracie sudah menempati ruangan 1!
Silakan cari ruangan lain untuk dokter dina.
```

Gambar 7.26 Ruangan Sudah Ditempati

#Kasus 4

Ruangan sudah ditempati dan dokter sudah diassign

```
Pilihan Anda: 11
>>> ASSIGN_DOKTER
Username: dina
ID Ruangan : 1

Dokter dina sudah ada di ruangan 2!
Ruangan 1 juga sudah ditempati dokter gracie!
```

Gambar 7.27 Ruangan Sudah Ditempati dan Dokter Sudah Diassign

12. F11 - Diagnosis

#Kasus 1

Berhasil diagnosis

```
Halo, DOKTER jack!
-----
Menu yang Tersedia:
1. LIHAT DENAH
2. LIHAT RUANGAN
8. DIAGNOSIS PASIEN
9. BERI PENGOBATAN
98. HELP
99. LOGOUT
0. EXIT
-----

Pilihan Anda: 8
>>> DIAGNOSIS
mi terdiagnosa penyakit Influenza!
```

Gambar 7.28 Berhasil Diagnosis Pasien

#Kasus 3

Tidak berhasil diagnosis karena input PASIEN tidak sesuai range penyakit apa pun

```
Pilihan Anda: 11
>>> DAFTAR CHECK-UP

Silakan masukkan data check-up Anda:
Suhu Tubuh (Celcius): 30
Tekanan Darah (Sistol/Diastol, contoh 120 80): 120 90
Detak Jantung (bpm): 0
Input tidak valid. Silakan masukkan angka positif.
Detak Jantung (bpm): 1
Saturasi Oksigen (%): 1
Kadar Gula Darah (mg/dL): 1
Berat Badan (kg): 1
Tinggi Badan (cm): 1
Kadar Kolesterol Total (mg/dL): 1
Kadar Kolesterol LDL (mg/dL): 1
Trombosit (ribu/ $\mu$ L): 1

Berikut adalah daftar dokter yang tersedia:
1. Dr. jack - Spesialisasi Umum - Ruangan 1 (Antrian: 1 orang)
2. Dr. andrew - Spesialisasi Umum - Ruangan 2 (Antrian: 0 orang)

Pilih dokter (1-2): 2

Pendaftaran check-up berhasil!
Anda telah terdaftar untuk check-up dengan Dr. andrew di Ruangan 2.
Posisi antrian Anda: 1
```

Halo, DOKTER andrew!

Menu yang Tersedia:

1. LIHAT DENAH
 2. LIHAT RUANGAN
 8. DIAGNOSIS PASIEN
 9. BERI PENGOBATAN
 98. HELP
 99. LOGOUT
 0. EXIT
-

Pilihan Anda: 8

>>> DIAGNOSIS

dif tidak terdiagnosis penyakit apapun!

Gambar 7.29 Pasien Tidak Terdiagnosis

13. F12 - Ngobatin

#Kasus 1

Berhasil ngobatin

```
Pilihan Anda: 9
>>> BEROBAT
Dokter sedang mengobati pasien!
Pasien mi memiliki penyakit Influenza.
Obat yang harus diberikan:
  1. Oseltamivir
  2. Vitamin C
Pasien mi telah selesai berobat.
```

Gambar 7.30 Pasien Berhasil Diobatin

#Kasus 2

Tidak berhasil ngobatin

```
Pilihan Anda: 9
>>> BEROBAT
Dokter sedang mengobati pasien!
Pasien tidak memiliki penyakit!
Pasien belum di diagnosis!
Pasien dif tidak memiliki penyakit yang terdiagnosis.
```

Gambar 7.31 Pasien Tidak Berhasil Diobatin

14. F13 - Aku boleh pulang ga, dok?

#Kasus 1

Berhasil pulang

```
Pilihan Anda: 10
>>> PULANGDOK

Dokter sedang memeriksa keadaanmu...

Selamat! Kamu sudah dinyatakan sembuh oleh dokter. Silahkan pulang dan semoga sehat selalu!
```

```

-----
Pilihan Anda: 13
>>> MINUM_OBAT
===== DAFTAR OBAT DI INVENTORY =====
1. Oseltamivir
2. Vitamin C
=====
Pilih obat untuk diminum (masukkan nomor): 1
GLEKGLEKGLEK... Oseltamivir berhasil diminum!!!

Sisa obat di inventory:
1. Vitamin C
=====

```

```

Pilihan Anda: 13
>>> MINUM_OBAT
===== DAFTAR OBAT DI INVENTORY =====
1. Vitamin C
=====
Pilih obat untuk diminum (masukkan nomor): 1
GLEKGLEKGLEK... Vitamin C berhasil diminum!!!

Sisa obat di inventory:
Inventaris obat Anda sekarang kosong.
=====

```

Gambar 7.32 Pasien Berhasil Pulang

#Kasus 2

Tidak berhasil pulang

```

-----
Pilihan Anda: 10
>>> PULANGDOK

Dokter sedang memeriksa keadaanmu...
Maaf, tapi kamu belum minum obat sesuai resep!

Urutan peminuman obat yang diharapkan:
Oseltamivir ->
Vitamin C

Urutan obat yang kamu minum:
Oseltamivir -> Vitamin C

Silahkan kunjungi dokter untuk meminta penawaran yang sesuai!

```

Gambar 7.33 Pasien Tidak Berhasil Pulang

15. F14 - Daftar Check-Up

#Kasus 1

Berhasil daftar check up

```

Pilihan Anda: 11
>>> DAFTAR CHECK-UP

Silakan masukkan data check-up Anda:
Suhu Tubuh (Celcius): -30
Input tidak valid. Silakan masukkan angka positif.
Suhu Tubuh (Celcius): 36
Tekanan Darah (Sistol/Diastol, contoh 120 80): 100
Input tidak valid. Silakan masukkan dua angka (sistol diastol).
Tekanan Darah (Sistol/Diastol, contoh 120 80): 100 70
Detak Jantung (bpm): 65
Saturasi Oksigen (%): 96
Kadar Gula Darah (mg/dL): 100
Berat Badan (kg): 50
Tinggi Badan (cm): 150
Kadar Kolesterol Total (mg/dL): 150
Kadar Kolesterol LDL (mg/dL): 150
Trombosit (ribu/μL): 150

Berikut adalah daftar dokter yang tersedia:
1. Dr. Jack - Spesialisasi Umum - Ruangan 1 (Antrian: 0 orang)

Pilih dokter (1-1): 1

Pendaftaran check-up berhasil!
Anda telah terdaftar untuk check-up dengan Dr. Jack di Ruangan 1.
Posisi antrian Anda: 1

```

Gambar 7.34 Pasien Berhasil Daftar Check Up

#Kasus 2

Sudah pernah daftar check up

```

Pilihan Anda: 11
>>> DAFTAR CHECK-UP

Anda sudah terdaftar dalam antrian check-up!
Silakan selesaikan check-up yang sudah terdaftar terlebih dahulu.

```

Gambar 7.35 Pasien Sudah Pernah Daftar Check Up

16. F15 - Antrian Saya!

#Kasus 1

Pasien berada di antrian

```

Pilihan Anda: 12
>>> ANTRIAN

Status antrian Anda:
Dokter: Dr. Francis Galton
Ruangan: 1
Posisi antrian: 1 dari 2

```

Gambar 7.36 Antrian Saya! (Ruangan penuh, antrian di luar ruangan)

#Kasus 2

Pasien belum melakukan command DAFTAR CHECK UP

```
Pilihan Anda: 12
>>> ANTRIAN

Anda belum terdaftar dalam antrian check-up!
Silakan daftar terlebih dahulu dengan command DAFTAR_CHECKUP.
```

Gambar 7.37 Antrian Saya! (Belum melakukan DAFTAR_CHECKUP)

#Kasus 3

Pasien berada di ruangan

```
Pilihan Anda: 12
>>> ANTRIAN

Status antrian Anda:
Dokter: Dr. jack
Ruangan: 1
Anda sedang berada di dalam ruangan dokter!
```

Gambar 7.38 Antrian Saya! (Sedang berada di ruangan dokter)

17. F16 - Minum Obat

#Kasus 1

Pilihan Invalid, tidak sesuai dengan jumlah obat di Inventaris

```
-----
Pilihan Anda: 13
>>> MINUM_OBAT
===== DAFTAR OBAT DI INVENTORY =====
1. Oseltamivir
2. Vitamin C
=====
Pilih obat untuk diminum (masukkan nomor): 4
Pilihan nomor tidak tersedia!
```

Gambar 7.39 Minum Obat (Pilihan Invalid)

#Kasus 2

Ada obat di inventory (dalam uji coba kali ini merupakan obat untuk penyakit Influenza)

```

-----
Pilihan Anda: 13
>>> MINUM_OBAT
===== DAFTAR OBAT DI INVENTORY =====
1. Oseltamivir
2. Vitamin C
=====
Pilih obat untuk diminum (masukkan nomor): 1
GLEKGLEKGLEK... Oseltamivir berhasil diminum!!!

```

Gambar 7.40 Minum Obat (Inventaris Pasien Isi)

#Kasus 3

Pasien belum mendapatkan obat untuk diagnosanya atau obat sudah diminum semua sehingga *inventory* kosong.

```

-----
Pilihan Anda: 13
>>> MINUM_OBAT
Inventaris obat Anda kosong! Tidak ada obat untuk diminum.

```

Gambar 7.41 Minum Obat (Inventaris Pasien Kosong)

18. F17 - Minum Penawar

#Kasus 1

Pasien belum meminum obat apapun dan stack perut kosong.

```

-----
Pilihan Anda: 14
>>> PENAWAR
Uwekkk!!! Oseltamivir keluar dan kembali ke inventory

```

Gambar 7.42 Minum Penawar (Stack IsiPerut Isi)

#Kasus 2

Obat yang telah diminum dimuntahkan kembali

```
-----  
Pilihan Anda: 14  
>>> PENAWAR  
Perut kosong!! Belum ada obat yang dimakan.
```

Gambar 7.43 Minum Penawar (Stack IsiPerut Kosong)

19. F18 - Exit

```
-----  
Pilihan Anda: 0  
Apakah Anda sudah selesai menggunakan layanan ini? (y/n) y  
Terima kasih telah menggunakan sistem rumah sakit Nimons. Sampai jumpa!
```

Gambar 7.44 Exit

20. B03 - Aura

```
Pilihan Anda: 11  
>>> DAFTAR CHECK-UP  
  
Silakan masukkan data check-up Anda:  
Suhu Tubuh (Celcius): 39  
Tekanan Darah (Sistol/Diastol, contoh 120 80): 125 76  
Detak Jantung (bpm): 110  
Saturasi Oksigen (%): 90  
Kadar Gula Darah (mg/dL): 175  
Berat Badan (kg): 60  
Tinggi Badan (cm): 150  
Kadar Kolesterol Total (mg/dL): 150  
Kadar Kolesterol LDL (mg/dL): 150  
Trombosit (ribu/ $\mu$ L): 150  
  
Berikut adalah daftar dokter yang tersedia:  
1. Dr. nic - Spesialisasi Umum - Ruangan 1 (Antrian: 0 orang) - Aura 0  
2. Dr. pedro - Spesialisasi Umum - Ruangan 2 (Antrian: 0 orang) - Aura 0
```

Pilihan Anda: 32

Sort berdasarkan?

1. ID
2. Username
3. Aura

>>> 3

Urutan?

1. ASC
2. DESC

>>> 1

>>> LIHAT DOKTER

ID	Username	Role	Aura
3	pedro	Dokter	0
2	nic	Dokter	1

Pilihan Anda: 11

>>> DAFTAR CHECK-UP

Silakan masukkan data check-up Anda:

Suhu Tubuh (Celcius): 36

Tekanan Darah (Sistol/Diastol, contoh 120 80): 100 70

Detak Jantung (bpm): 65

Saturasi Oksigen (%): 96

Kadar Gula Darah (mg/dL): 100

Berat Badan (kg): 55

Tinggi Badan (cm): 150

Kadar Kolesterol Total (mg/dL): 150

Kadar Kolesterol LDL (mg/dL): 150

Trombosit (ribu/ μ L): 150

Berikut adalah daftar dokter yang tersedia:

1. Dr. nic - Spesialisasi Umum - Ruangan 1 (Antrian: 0 orang) - Aura 1
2. Dr. pedro - Spesialisasi Umum - Ruangan 2 (Antrian: 0 orang) - Aura 0

Gambar 7.45 Aura

LAMPIRAN

[Form Asistensi](#)